



**UNIVERSITÀ
DEGLI STUDI
DI TRIESTE**

UNIVERSITÀ DEGLI STUDI DI TRIESTE

XXXVII CICLO DEL DOTTORATO DI RICERCA IN

APPLIED DATA SCIENCE AND ARTIFICIAL INTELLIGENCE

**COMMUNITY DETECTION IN COMPLEX NETWORKS:
LIMITATIONS AND NOVEL APPROACHES
TO ENHANCE SOLUTION STABILITY**

Settore scientifico-disciplinare: SECS-S/05 STATISTICA SOCIALE

**DOTTORANDO
FABIO MOREA**

**COORDINATORE
PROF. FRANCESCO PAULI**

**SUPERVISORE DI TESI
PROF. DOMENICO DE STEFANO**

ANNO ACCADEMICO 2023/2024

Dedico questa tesi a Valentina, Lorenzo ed Elena, che mi hanno supportato e incoraggiato con amore e pazienza in questi tre anni di studio.

Contents

Abstract	vii
Abstract (italiano)	ix
Acronyms and symbols	xi
Introduction	1
1 Innovation	5
1.1 Knowledge diffusion and innovation	6
1.2 Collaboration as a driver for innovation	7
1.3 Innovation Networks	8
2 Fundamentals of Network Analysis	11
2.1 Network definition and notation	11
2.2 Centrality Measures	13
2.3 Components	14
2.4 Communities	15
2.5 Partitions	16
2.6 Temporal Network Analysis in Longitudinal Data	18
2.7 Community detection algorithms	19
2.8 Benchmark networks	21
3 Limitations of community detection	25
3.1 Variability	25
3.2 Validity of results	27
3.3 Outliers	29
3.4 Input ordering bias	30
4 Enhancing stability of community detection	33
4.1 Solution Space	34
4.2 Consensus Community Detection	46
4.3 Performance of CCD	53
5 R-package ‘communities’	61
5.1 Functions to generate benchmark networks	61

5.2	Functions for solution space and quality check	63
5.3	Functions for consensus community detection	66
5.4	Functions for analysing community structure	68
5.5	Functions for visualization	69
6	Innovation patterns within a regional economy	71
6.1	Innovation in Friuli Venezia Giulia region	72
6.2	Data and methodology	73
6.3	Results and discussion	75
7	Mapping leadership and communities in EU-funded research	87
7.1	Horizon programmes	87
7.2	The 'hydrogen energy' sector and NAHV project	88
7.3	Data acquisition	90
7.4	Data preparation	91
7.5	Results	92
	Conclusions	101
A	Appendix: Open Science	105
A.1	FAIR principles	105
A.2	Open publication	106
A.3	Open access to data	107
A.4	Open access to code and software	108
	Acknowledgments	110
	Bibliography	111

Abstract

This thesis presents the research conducted as part of the PhD program in *Applied Data Science and Artificial Intelligence*, to establish a methodological framework for identifying, measuring, and interpreting collaborative dynamics that drive innovation, through the application of network analysis.

Collaboration among businesses, research centres, and policymakers is a core process in innovation; however, the lack of structured data and the heterogeneity of interactions make this study particularly challenging. The proposed methodology is applied to datasets that describe relationships among entities (such as companies, organizations, or individuals) collaborating over time. The data are segmented into temporal intervals and used to represent collaborations as weighted networks. By applying standard techniques like centrality measures, influential actors can be identified, while community detection algorithms reveal cohesive groups. Comparing networks across consecutive time intervals provides insight into the evolution of collaborations.

The methodology and tools developed were applied to two case studies to demonstrate their practical relevance. The first investigates innovation dynamics in the Friuli Venezia Giulia region, emphasizing the interactions among industries, universities, and research centres as drivers of regional innovation. The second explores patterns of leadership and collaboration within EU-funded research projects, with a particular focus on the hydrogen energy sector. These applications highlight the effectiveness of the proposed approach in extracting meaningful insights from collaborative networks and addressing concrete challenges in data-driven innovation studies.

Solving this applied problem required a novel theoretical and methodological development, which became an integral and distinctive component of the thesis. Community detection introduces several well-known challenges, such as result variability, the need for validation, and sensitivity to input data ordering. The objective was to achieve a solution that is stable (i.e. minimizing dependence on stochastic factors) while being able to manage the fuzziness of collaborations and the presence of outliers.

The solution was found by developing an innovative paradigm, based on the idea that community detection algorithms generate a single point within a broader solution space. This solution space must therefore be generated (through an optimized process) and subsequently analysed. The methodology involves generating multiple solutions until a sufficiently stable solution space is detected, followed, whenever necessary, by a consensus procedure to obtain a single result. The process produces a node-level uncertainty coefficient and introduces strategies to effectively manage outliers, enhancing the interpretability and reliability of the outcomes.

One of the key outputs of this thesis is the `communities` R package, which provides the complete algorithm (solution space exploration, consensus community

detection, and node-level uncertainty quantification). The package facilitates reproducibility, allowing anyone to assess the robustness and reliability of results. The open-source nature of the software aligns with FAIR principles (Findable, Accessible, Interoperable, Reusable), promoting transparency and accessibility in network research.

The proposed methodological contribution extends beyond the specific context of this thesis and can be applied to other domains of network analysis, offering a versatile tool for analysing complex and dynamic collaborative phenomena.

Abstract (italiano)

Questa tesi illustra il lavoro di ricerca svolto nel contesto del programma di dottorato in Applied Data Science and Artificial Intelligence, con l'obiettivo di definire un quadro metodologico per l'identificazione, la misurazione e l'interpretazione delle dinamiche collaborative che promuovono l'innovazione, utilizzando metodi di network analysis.

La collaborazione tra imprese, centri di ricerca e decisori politici rappresenta un elemento cruciale per l'innovazione; tuttavia, la mancanza di dati strutturati e l'eterogeneità delle interazioni rendono lo studio particolarmente complesso. La metodologia proposta si applica a dataset che descrivono una relazione tra soggetti (imprese, centri di ricerca, organizzazioni, persone...) che collaborano nel tempo. I dati vengono segmentati in intervalli temporali ed utilizzati per rappresentare le collaborazioni come reti pesate. Applicando tecniche standard come le 'misure di centralità' è possibile individuare i soggetti più influenti, mentre gli algoritmi di 'community detection' consentono di identificare gruppi coesi. Il confronto tra le reti in successivi intervalli di tempo offre una visione dell'evoluzione delle collaborazioni.

La metodologia e gli strumenti sviluppati sono stati applicati a due casi di studio per dimostrarne le applicazioni pratiche. Il primo esamina le dinamiche dell'innovazione nella regione Friuli Venezia Giulia, evidenziando l'interazione tra industrie, università e centri di ricerca come motore dell'innovazione regionale. Il secondo esplora i modelli di leadership e collaborazione nei progetti di ricerca finanziati dall'Unione Europea, con un focus sul settore dell'energia da idrogeno. Queste applicazioni dimostrano l'efficacia dell'approccio proposto nell'identificare informazioni significative all'interno delle reti di collaborazione e nell'affrontare sfide concrete negli studi sull'innovazione guidata dai dati.

La soluzione di questo problema applicativo ha richiesto un approfondimento teorico e metodologico che ha dato origine ad una componente importante ed aggiuntiva della tesi. L'identificazione delle comunità, infatti, introduce problematiche ampiamente riconosciute in letteratura, come la variabilità dei risultati, la necessità di validazione e la sensibilità all'ordine dei dati in ingresso. L'esigenza è ottenere una soluzione stabile (che riduca al minimo la dipendenza da fattori stocastici) e al tempo stesso capace di gestire efficacemente la complessità delle relazioni e la presenza di outlier.

La soluzione è stata trovata definendo un paradigma innovativo, basato sull'idea che gli algoritmi di rilevazione delle comunità generano un singolo punto all'interno di un più ampio *spazio delle soluzioni*. Lo spazio delle soluzioni deve quindi essere generato (con un processo ottimizzato) e successivamente analizzato. La metodologia prevede la generazione di soluzioni multiple, fino a raggiungere un quadro sufficientemente stabile, seguito quando necessario, dall'applicazione di una procedura di consenso per raggiungere un risultato univoco. Il processo genera un

coefficiente di incertezza a livello di nodo e introduce strategie per gestire efficacemente gli outlier, migliorando l'interpretabilità e l'affidabilità del risultato.

Uno dei prodotti di questa tesi è il pacchetto R `communities`, che mette a disposizione l'algoritmo completo (esplorazione dello spazio delle soluzioni, consensus community detection e la quantificazione dell'incertezza a livello di nodo). Il pacchetto facilita la riproducibilità dei risultati, consentendo a chiunque di valutare l'affidabilità e la robustezza dei risultati. La natura open-source del software è allineata ai principi FAIR (Findable, Accessible, Interoperable, Reusable), promuovendo trasparenza e accessibilità nella ricerca sulle reti.

Il contributo metodologico proposto non si limita al contesto specifico della tesi, ma può essere esteso ad altri settori della network analysis, offrendo uno strumento versatile per analizzare fenomeni collaborativi complessi e dinamici.

Acronyms and symbols

The following is a comprehensive list of acronyms and symbols used in this document, presented in the order in which they first appear in the text. Each entry includes a brief description to clarify its meaning and usage, ensuring consistency and ease of reference throughout the work.

Chapter 2

G	A network, i.e. a set of vertices, edges and weights.
V	The set of vertices of G
u, v	Nodes of the network
n_v	the number of nodes in the network $n_v = V $
E	The set of edges connecting pair of nodes in G .
n_e	the number of edges in the network $n_e = E $
W	The weights associated to the edges
\mathbf{A}	the adjacency matrix of a one-mode network
\mathbf{B}	the adjacency matrix of a two-mode network.
\mathcal{K}	A component of the network: $\mathcal{K} \subseteq G$.
\mathcal{A}	A community detection algorithm: $\mathcal{A}(G, \rho) \rightarrow \mathcal{P}$
ρ	The parameter(s) of the community detection algorithm
l_i	Membership label mapping node v to the community C_i
C_i	A community within the network: $C_i \subseteq V$, as a list of pairs (<i>node, label</i>)
IM.....	Infomap algorithm
LD.....	Leiden algorithm

LV.....	Louvain algorithm
LP.....	Label Propagation algorithm
WT.....	Walktrap algorithm
\mathcal{P}	A partition of the network, i.e. a set of communities: $\mathcal{P} = \{C_1, \dots, C_k\}$.
k	the number of communities that compose a partition \mathcal{P}
Q	Modularity of a partition
μ	Mixing parameter of a partition
\mathcal{G}_t	A family of networks, segmented by time intervals.
RC.....	Ring of Cliques, an artificial benchmark network.
LFR.....	Lancichinetti – Fortunato – Radicchi: an artificial benchmark network

Chapter 3

G^*	a random permutation of edges and vertices of G
\mathcal{S}	The solution space generated by \mathcal{A} after t trials.
t	the number of trials used to generate the solution space
t_{max}	the maximum number of trials
ns	the number of partitions that compose \mathbf{S} i.e., $ns = \mathbf{S} $
CCD.....	Consensus Community Detection
γ	uncertainty coefficient associated to each node
\widetilde{C}_i	An extended community represented by a list of triplets (<i>node, label, γ</i>)
$\widetilde{\mathcal{P}}$	An extended partition: $\widetilde{\mathcal{P}} = \{\widetilde{C}_1, \dots, \widetilde{C}_k\}$.

Chapter 4

LFR.....	A set of artificial benchmark networks.
k_0	number of cliques in the RC
s_0	size of each clique in the RC

Chapter 6

FVG.....	Friuli Venezia Giulia region
LMN-FVG.	Labour Market Network in Friuli Venezia Giulia
SiS FVG...	Scientific and Innovation System of Friuli Venezia Giulia
SME.....	Small and Medium sized Enterprises International Standard
EU.....	European Union
EIS.....	European Innovation Scoreboard
ISCO.....	International Standard Classification of Occupations

Chapter 7

CORDIS... Community Research and Development Information Service: the Open Data source on Horizon Programmes supported by the European Commission.

NAHV North Adriatic Hydrogen Valley

\mathcal{G}_y A family of networks segmented by year.



Introduction

This thesis presents the research conducted as part of a PhD program in Applied Data Science and Artificial Intelligence. The central theme of this work revolves around advancing the field of community detection within complex networks, specifically addressing the challenges related to the stability and reproducibility of the detected communities.

Innovation, as a fundamental strategy for organizations and policymakers, provides a compelling framework to motivate this research. Collaboration between academia, industry, government and society is essential for achieving technological advancements and fostering creative solutions to complex problems. Understanding the structure and dynamics of the collaborations is therefore relevant for policymakers addressing social and economic issues, as well as for organizations operating in competitive environments.

However, collaborations are difficult to study due to the lack of structured and comprehensive data, and the deep dynamics of innovation remain elusive.

The proposed approach is to model collaboration as networks, identify leading organizations and cohesive groups, and track their evolution over time.

The workflow begins by preparing and segmenting the data into relevant time slices to enable temporal analysis of collaboration patterns. For each time slice, collaborations between organizations are represented as weighted networks that reflect the strength of their relationships. Network analysis techniques are then employed: centrality measures are used to identify leading organizations, and community detection methods reveal cohesive groups of collaborators. By comparing networks across time, the evolution of collaboration dynamics can be observed.

However, testing this approach on both artificial and real-world networks revealed several issues. Many algorithms, when applied to complex cases, produced invalid or inconsistent results, with outcomes changing at each run.

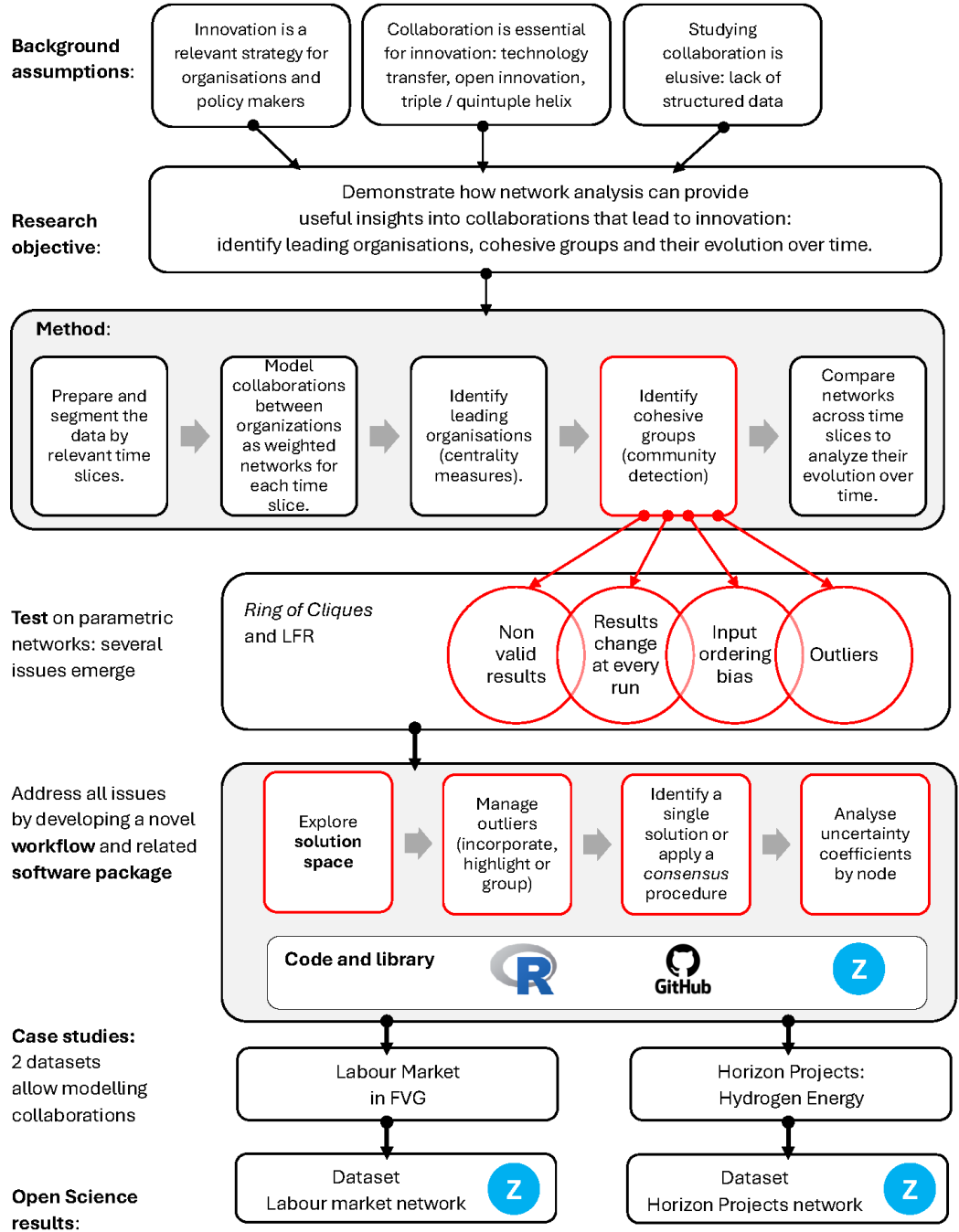
Acknowledging variability prompts a fundamental paradigm shift in community detection. Algorithms should not be viewed as delivering a single, definitive solution. Instead, each run reveals only one possible outcome within a larger *solution space*. Through repeated execution, a clearer understanding of the solution space's topology can gradually emerge.

To explore this topic further, a specific parametric network called the "Ring of Cliques" was developed, capable of generating different communities ranging from trivially simple to extremely fuzzy. This approach helped identify input-ordering bias and led to the development of a strategy for managing outliers.

In response to these challenges, an **extended workflow** were created. This novel workflow introduces the concepts of exploring the solution space, which can have a single, dominant or multiple solutions. In the latter case, a *consensus* procedure can be applied to deliver a stable solution. Such procedure is derived by other proposed in literature, and enriched by a more nuanced management of the prevalence of different solution in the solution space, Moreover the consensus approach also generates "uncertainty coefficients" for each node, which offer a more robust understanding of the network structure. Finally, the workflow introduces a taxonomy to manage outliers, with three cases: incorporating, highlighting, or grouping. The overall performance of the new approach has been tested on artificial benchmark networks, and resulted in the creation of the **communities R package**, which is openly available on GitHub and Zenodo.

Two **case studies** are presented, based on relevant and largely unexplored datasets. The first is the Labour Market Network in Friuli Venezia Giulia region, focusing on the movement of workers across various types of organizations—such as academic institutions, research centers, and industries—as a proxy for the exchange of knowledge, skills, and innovation. The second is the data on Horizon 2020 and Horizon Europe programs, focusing on the 'hydrogen energy' sector. These case studies demonstrate the effectiveness of network analysis in revealing collaboration dynamics and contribute to the growing body of open science resources.

visual summary



Z = published in Zenodo

GitHub = development version published in GitHub

Innovation

Innovation is the process of creating new value by introducing new products, services or methods that benefit customers or society at large [81, 80]. This process is crucial for societal development, as it facilitates adaptation to changing environments and the resolution of emerging challenges. Innovation plays a particularly significant role in research-industry collaborations, where academic research provides cutting-edge advancements, while industry brings knowledge of market demands and practical applications. Together, these efforts drive economic growth and promote societal progress.

In **public research institutions**, such as universities and research centres, innovation is central to their mission of advancing knowledge and making tangible contributions to industry and society. These organizations strive not only to explore theoretical knowledge but also to apply their findings to real-world problems. By engaging in innovation, they bridge the gap between research and industry, ensuring that their discoveries contribute to technological advances, economic productivity, and social progress. This collaborative effort often results in the commercialization of research, where academic findings are translated into products, services, or solutions that benefit both the public and private sectors.

For industrial **companies**, innovation is crucial for maintaining competitiveness in a fast-paced global market. Companies that continuously innovate are better positioned to differentiate themselves from their competitors by offering new and improved products or services. In industries where consumer needs and preferences rapidly evolve, innovation becomes a key driver of success. It allows businesses to adapt, meet market demands, and provide superior value to their customers. In this way, innovation not only fosters growth but also strengthens a company's market position and long-term sustainability.

1.1 Knowledge diffusion and innovation

The concept of innovation is well rooted in literature, dating back to Joseph Schumpeter's work from the 1930s [86], when it has been recognized as a key driver of economic growth and transformation. Initially, innovation was seen as confined within company boundaries and driven by in-house research and development. However, with globalization and technological advances, this "closed" approach became limiting.

Technology transfer is a model of collaboration for innovation that involves the formal exchange of knowledge, technologies, or expertise between organizations, typically from research institutions to commercial entities. This process is often structured through contracts, such as licensing agreements, patents, or joint ventures, where intellectual property is transferred under defined legal terms. Technology transfer is a relevant driver of innovation, as it allows scientific advancements to be commercialized and applied in real-world industries, leading to new products, processes, and services.

While technology transfer is a valuable driver of innovation by facilitating the commercialization of scientific advancements, it alone is not sufficient.

Open innovation, introduced by Chesbrough, is "a distributed innovation process based on purposively managed knowledge flows across organizational boundaries" [16]. In this a paradigm companies integrate external ideas and technologies with internal ones to enhance their products and services. This approach contrasts with the traditional "closed" innovation model, which relies solely on in-house research and development. Open innovation promotes collaboration with external partners, such as universities, research institutions, suppliers, and customers, to access a broader range of expertise and resources. This can accelerate product development, reduce costs, and provide access to new markets and technologies.

Since its introduction, the open innovation model has continued to evolve, adapting to new technological advancements and shifting business environments, and it remains a vital and valid concept for interpreting innovation today.

Bertello et al. [6] offer a more recent analysis of open innovation, examining its research evolution over the past decade using bibliometric techniques and content analysis to explore the field's knowledge structure and theoretical developments. Also, the Economist Impact's project on open innovation [45] provides an in-depth analysis of how the open innovation model has adapted to the evolving technological and business landscape. It emphasizes the importance of cross-sector collaboration and the role of digital transformation in facilitating the flows of knowledge between an organisation and the external expertise from universities, research institutions, suppliers, and customers.

Similarly, the Open Innovation Briefing Paper [46] emphasizes the critical role of collaboration in driving innovation. It outlines key benefits such as integrating diverse expertise, resource sharing, speeding up development, and accessing new markets. These collaborative efforts foster co-creation and the exchange of ideas, enabling faster innovation and cost reduction.

In the open innovation paradigm, **inbound and outbound knowledge flows** are essential, enabling a company to acquire new ideas, or to exploit a market opportunity to support innovation within a wider value chain.

1.2 Collaboration as a driver for innovation

As noted by Meireles, Azevedo, and Boaventura [87], collaboration beyond internal research and development, creates networks that facilitate the exchange of crucial knowledge and resources.

Collaboration drives innovation at multiple levels. *Within* organizations, different departments collaborate to integrate diverse expertise. *Between* organizations, partnerships enable the sharing of complementary assets, such as technology and market access. Beyond the private sector, collaboration with public institutions, non-profit organisations and governments addresses broader societal challenges, like sustainability and public health. Overall, collaboration allows to co-create value (i.e. to innovate), solving complex problems that would be unachievable in isolation.

The open innovation model has thus evolved to include frameworks such as the triple helix and quadruple/quintuple helix models. The triple helix model [26] highlights the synergy between universities, industry, and government in fostering innovation. The quadruple and quintuple helix models [12] build on this by incorporating civil society and environmental sustainability into the innovation process.

As Meireles et al. argue, the existing literature offers valuable insights but lacks a comprehensive, data-driven framework for understanding large-scale collaboration. Measuring the relation between collaboration and innovation remains challenging, primarily due to the scarcity of comprehensive data. Collaboration often happens through informal networks, reserved contracts or ad hoc partnerships, which are not consistently tracked. consequently, much of the available information is either restricted, confidential, or anecdotal.

Additionally, the absence of standardized metrics for innovation complicates efforts to assess collaboration. Innovation outcomes vary widely — some bring immediate financial returns, while others offer long-term strategic value, such as enhancing a company's reputation or market position. This diversity makes it

hard to quantify collaboration's role and impact on innovation. This thesis aims to address these issues, by developing a methodology to capture the existence and measure the intensity of collaborations, at least in some domains where structured datasets are available.

From the perspective of individual **organizations**, understanding the dynamics of innovation at the regional or international level can be a good opportunity. Consider, for example, an organization involved in a new hydrogen valley project: it must not only focus on the development of its own project activities, but it may benefit from identifying partners and competitors that are engaged in other hydrogen valleys. This broader awareness may help identifying synergies, risks, and market opportunities.

Policymakers adopt yet another perspective. They have a strategic vision to align local, national, and European resources, aiming for medium- and long-term impact of the innovation ecosystems. Achieving this requires a clear understanding of how collaborations emerge between industry and research, who the key stakeholders are, and how these collaborations evolve over time.

1.3 Innovation Networks

Networks provide a powerful framework for modeling collaboration by representing knowledge exchange between organizations. In these models, nodes typically represent organizations, while edges represent the collaborative effort or knowledge exchange between them. Identifying nodes is straightforward, but defining the edges is more complex, as the modeling choices are heavily influenced by the available data—or the lack thereof. The type of data available determines which aspects of the relationships between nodes can be represented, how accurately and under which time frame.

The key challenge is selecting the appropriate data to accurately reflect the nature of collaboration, which directly influences our understanding of innovation dynamics.

This thesis explores the use of networks as models for collaboration-driven innovation, addressing three key research questions:

1. Which organizations are the most influential in establishing collaborations, and driving knowledge flows?
2. Are there any meaningful communities within the network?
3. How do knowledge flows, leading roles and communities evolve over time?

These questions are relevant to two main stakeholder groups: project managers and policymakers. Project managers are interested in understanding their organization's position within a broader network, identifying sector leaders, and engaging

with the primary knowledge flows. Policymakers, on the other hand, aim to evaluate whether their regional policies have enhanced leadership roles over time and whether knowledge flows link organizations in their territories with more innovative and successful counterparts.

Key requirements of this approach are that the results must be independent of contingent factors (such as software implementation or ordering of the input data) and tested for validity. Additionally, if multiple algorithmic options are available, the selection must be data-driven and performance-oriented, ensuring that the chosen algorithm yields the most interpretable and reliable outcomes.

Fundamentals of Network Analysis

This chapter introduces the definitions and notation that will be used throughout the thesis. It covers fundamental concepts in network analysis, including centrality measures, partitions, community detection algorithms and temporal analysis. The examples, Whenever possible, will refer to the context of innovation networks illustrated in the previous chapter.

2.1 Network definition and notation

A graph, or network, is defined as a set $G = \{V, E, W\}$ where V is the set of vertices (nodes), E is the set of edges connecting pairs of nodes and W is the set of weights corresponding to the elements of E . The number of vertices in the graph is $n_v = |V|$ and the number of edges and weights is $n_e = |E|$.

Formally, V and E are sets, i.e. there is no inherent order in the arrangement of nodes and edges. However, in practice, the representation of a network in a file or in a data structure in a programming language is inherently ordered, taking the form of a data frame or a matrix. This implies that network analysis algorithms may suffer of biases based on the order in which nodes and edges are processed. The potential impact of such input order biases will be explored in detail in Section 3.4.

2.1.a Weighted and unweighted networks

The first step in network analysis is constructing the network by identifying its nodes, edges, and weights. Networks are typically derived from tabular data, which consists of a list of nodes and a corresponding list of edges—pairs of connected nodes. A more compact way to represent this information is through matrix.

The *adjacency matrix* is a mathematical representation of a network that describes the relationships between its nodes. For a network with n_v nodes, the adjacency matrix \mathbf{A} is a square matrix of size $n_v \times n_v$, where each element A_{ij} represents the connection between node i and node j . An unweighted matrix is represented by:

In an *unweighted* network, the elements of the matrix are binary, meaning:

$$\mathbf{A}_{ij} = \begin{cases} 1, & \text{if there is an edge between node } i \text{ and } j \\ 0, & \text{if no edge exists between node } i \text{ and } j \end{cases}$$

When not all connections are of equal importance - this is the case for all networks discussed in the case studies of this thesis - a **weighted** networks offer a more nuanced model. The elements of the adjacency matrix in a weighted network $\tilde{\mathbf{A}}_{ij}$ are represented by the weight of the edge connecting nodes i and j , or zero if no connection exists:

$$\tilde{\mathbf{A}}_{ij} = \begin{cases} \mathbf{W}_{ij}, & \text{if there is an edge between node } i \text{ and } j \\ 0, & \text{if no edge exists between node } i \text{ and } j \end{cases}$$

For example, in a network where edges represent knowledge transfer and collaboration-driven innovation, the weights could reflect the intensity or frequency of collaborative efforts between organizations. A higher weight on an edge between two companies might indicate frequent joint research projects, shared intellectual property, or significant knowledge exchange, all of which drive innovation. In contrast, a lower weight could represent occasional interactions or minimal knowledge sharing.

2.1.b Directed and undirected networks

Networks can be classified as either directed or undirected, depending on the nature of the relationships between nodes. In undirected networks, the edges represent mutual or bidirectional connections, meaning the order of nodes in each edge does not matter. These networks are often used to model symmetric relationships, such as friendships or collaborations. The adjacency matrix of an undirected network is symmetric.

In contrast, directed networks use edges where directionality is important, with edges represented as ordered pairs. These networks model asymmetric relationships, such as hierarchies or web links, where a connection from node u to node v does not necessarily imply a reciprocal connection from v to u .

2.1.c One-Mode or Two-Mode networks

The networks discussed thus far are one-mode networks, meaning all nodes are of the same type and belong to the same set V . In contrast, a two-mode network, also known as a bipartite network, has two distinct sets of nodes, denoted as V' and V'' , and edges can only connect a node in $u \in V'$ to a node in $v \in V''$.

For the purpose of this thesis, a one-mode network is a suitable model. However, in some cases two-mode networks are more easily derived from tabular data. For example in the case study presented in chapter 7 the construction of a two mode network is straightforward, with nodes in V' as projects, and nodes in V'' as organisations. In such case the matrix representation is in the form of rectangular matrix \mathbf{B} of size $|V'| \times |V''|$. A one-mode network can be obtained from a two-mode network via matrix product: $\mathbf{A} = \mathbf{B}^T \mathbf{B}$, where \mathbf{B}^T is the transposed of \mathbf{B} . Figure 2.1 illustrates the two-mode network of organisations by projects (left) and the one-mode network consisting only of organisation-by-organisation ties (right).

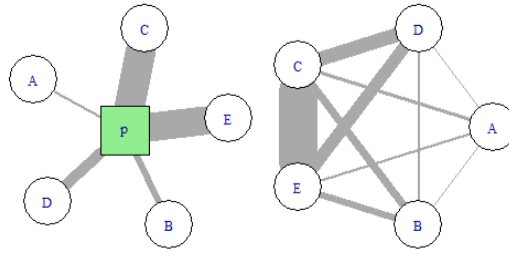


Figure 2.1: Schematic view of a project as two-mode network (left) and as one-mode network (right). Uppercase letters A, B, C, D, E represent organisations. The green square denoted with P represents a project. Edge width is proportional to the weight, i.e. the value of the contribution of each organisation to the project. The sum of weights in the two-mode network is equal to the sum of weights in the one-mode network.

2.2 Centrality Measures

The network structure can be evaluated using centrality measures such as degree, strength, and coreness to determine the importance and influence of nodes, each offering distinct insights.

The **degree** of a node v , denoted as $deg(v)$ is the simplest centrality measure, calculated as the number of edges incident to that node. In this context, the degree reflects the number of projects in which an organization is involved during a given year. A higher degree indicates that the node plays a more locally central role within the network, as it has a greater number of direct interactions with other nodes; in the context of our research, a node with a high degree represents an organisation that

is, or has been, a partner in many large projects, reflecting its extensive collaborative involvement.

Strength of a node is the sum of the weights of the edges incident to that node: $s(v) = \sum_{u \in N(v)} w_{vu}$ where $N(v)$ is the set of neighbours of v , and w_{vu} represents the weight of the edge between nodes v and u . In weighted networks, strength provides a more nuanced measure of a node's connectivity: with reference to Figure 2.1, all nodes have the same degree, but for example $s(C) \gg s(A)$. In this context, the strength reflects the monetary value of the projects in which an organization is involved during a given year.

The k -coreness (or **coreness**) [3] of a node is a measure of the node's position within the network's hierarchical structure, based on its connectivity. Specifically, a node has a k -coreness of k if it belongs to the k -core of the network. The k -core is a maximal subgraph in which every vertex has at least degree k , i.e. within this subgraph, each node is connected to at least k other nodes. In this context, coreness can be interpreted as the capacity of an organisation to partner with other organisations that, in turn, possess the same level of collaborative capacity.

Centrality measures are computed individually for each $G_y \in \mathcal{G}$ and saved as attributes of the nodes in G_y . This allows to track changes and compare the network structure across different years, providing insights into the dynamics of the collaborations and the shifting roles of organisations over time.

2.3 Components

In network theory, a **component** refers to a subset of the network where any two nodes are connected by a path, and no node in the subset is connected to any node outside of the component. Formally, a component K_i is a maximal subgraph of G such that all its nodes are connected internally and disconnected externally from other components. Let K_i denote the i -th component of a network G . Vertices are internally connected if

$$\forall u, v \in V_i, \exists \text{ a path from } u \text{ to } v \text{ in } K_i$$

and externally disconnected if

$$\forall u \in V_i, \forall v \in V_j, (u, v) \notin E$$

If a network consists of a single component, all nodes in the network are reachable from one another, either directly or through intermediate nodes. This is usually the case of simple networks, and artificial benchmark networks. However, a common feature in networks derived from real world data is the presence of a *giant compo-*

nent—a single, large subgraph that includes the majority of the nodes. In addition to this giant component, smaller, disconnected components may exist, though they typically contain significantly fewer nodes. An example of network with multiple components is provided in figure 6.2.

Analysing components provides insights into the connectivity of a network, helping to identify isolated groups that may function differently from the core network. In the context of innovation and knowledge transfer networks, the giant component is especially important because it represents the subset of nodes where most collaborations and exchanges of knowledge occur. Organizations within the giant component are usually highly interconnected, which facilitates the transfer of innovative ideas, research, and technologies. Conversely, organizations in smaller components are often less integrated into the knowledge network, potentially limiting their access to critical information and innovations.

2.4 Communities

A **community** is defined as a set of vertices $C_i \subseteq V$ that satisfies a condition: nodes that belong to C_i are *more densely connected* within each other than with the rest of nodes in V .

In practical terms, \mathbf{C} is a mapping each node $v \in V$ to a label $l \in \{l_1, l_2, \dots, l_k\}$ that identifies the community to which the node belongs. This mapping can be represented as a vector of pairs, where each pair consists of a *node* and its corresponding *label*:

$$\mathbf{C} = \begin{pmatrix} (v_1, l_1) \\ (v_2, l_2) \\ \vdots \\ (v_n, l_n) \end{pmatrix}$$

Here, v_i represents the i -th node, and l_i represents the label of the community to which the node is assigned. In a programming language as R or Python this structure is well represented by a data frame.

The expression *more densely connected* can be interpreted in various ways. In this thesis, it is assumed that the network is a one-mode, weighted and undirected, and the condition above is interpreted as follows. For a given community C_i , two subsets of edges are identified:

- E^{int} , which consists of edges connecting pairs of nodes within C_i ,
- E^{ext} , where each edge connects one node in C_i to a node in C_j , with $i \neq j$

The total weights associated with these edges are denoted as w_i^{int} and w_i^{ext} respectively. The community C_i is considered valid if the following condition holds:

$$\sum w_i^{\text{int}} > \sum w_i$$

It is important to note that, according to this definition, a community consisting of a single node (referred to as a **singleton**) necessarily has $w_i = 0 \forall i$, and thus cannot be considered a valid community under the criteria outlined above. This scenario frequently arises in real-world networks, where singletons may either be treated as exceptions to the rule, classified as outliers, or regarded as non-valid outputs.

Another implicit requirement for a community is that it should be **internally connected** i.e. given any pair of nodes $u, v \in C_i$, there must exist a path connecting them. While this condition may seem trivial, it is not always respected in practice. For instance, the Louvain (LV) algorithm described in 2.7 can produce communities that are internally disconnected, thereby violating this principle. This issue is particularly problematic because some tools, such as the `igraph` library, do not automatically check for or flag such cases. As a result, an algorithm may report a partition that includes communities failing to meet the connectivity criterion without any indication that the result is invalid. Therefore, it is crucial to perform a post-algorithm check to ensure the validity of the generated communities, as those described in 3.2.

2.5 Partitions

A **partition** \mathcal{P} is a set of disjoint communities whose union is equal to the whole graph. Formally:

$$\mathcal{P} = \{C_1, C_2, \dots, C_k\} \text{ such that } \begin{cases} C_1 \cup C_2 \cup \dots \cup C_k = V \\ C_i \cap C_j = \emptyset \quad \forall i \neq j \end{cases}$$

This implies that \mathcal{P} divides the network into non-overlapping groups, where each node belongs to exactly one group. While this definition is widely adopted in network analysis, other types of partitions are also possible. For instance, fuzzy partitions allow nodes to belong to multiple communities with varying degrees of membership. Similarly, partitions with overlapping nodes permit nodes to participate in more than one community. However, in this thesis, the focus will be exclusively on non-overlapping partitions, as they provide clearer analysis and interpretation.

2.5.a Mixing parameter

The fuzziness of a network partition \mathcal{P} can be measured using the **mixing parameter** μ defined as:

$$\mu = \frac{\sum_i d_i^{ext}}{\sum_i d_i^{total}}$$

where d_i^{ext} is the external degree of node i , which corresponds to the number of edges connecting node i to other nodes in different communities, and d_i^{total} is the total degree of node i . Consequently, μ takes values between 0 and 1. The mixing parameter, μ , takes low values in networks with well-defined community structures, where there are minimal connections between different communities.

The parameter μ has some similarities to the condition of validity we previously defined for individual communities. While the validity condition for communities uses the sum of the weights, here, for partitions, μ is based on the degree, or the count of edges, across the entire network.

2.5.b Modularity

Modularity is a widely used as an objective functions for community detection algorithms. It measures the quality of a partition by comparing the actual density of internal and external edges to the expected density in a randomized network with the same degree distribution. Formally, modularity Q is defined as:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{d_i d_j}{2m} \right] \delta(l_i, l_j)$$

where:

- A_{ij} is the element of the adjacency matrix representing the edge between nodes i and j ,
- d_i and d_j are the degrees of nodes i and j ,
- m is the total number of edges in the network,
- $\delta(l_i, l_j)$ is the Kronecker delta function that equals 1 if nodes $l_i = l_j$ (nodes i and j are in the same community), and 0 otherwise.

The modularity score Q ranges from -1 to 1 , where a higher value indicates a better division of the network into distinct communities. A value of Q close to 0 suggests no significant community structure, while a value closer to 1 implies that the partition captures well-defined and dense communities.

While modularity is a powerful tool for community detection, it has several limitations, which can affect the accuracy and reliability of its results. One issue

is the resolution limit, where small communities, especially in large networks, may be overlooked and merged into larger ones, even if they have meaningful structures. Moreover, modularity tends to favour balanced partitions, leading to misleading results in networks where the actual community sizes vary significantly. Another problem is *degeneracy*, which occurs when different partitions produce similar modularity scores, making it difficult to determine the optimal one partition. This is the basis for the variability of results discussed in chapter 3.1.

Ideally, algorithms should consistently produce a single, valid partition, but in practice, this is not always the case. The partition may violate the requirement of internal connectivity, or have a mixing parameter $\mu > 0.5$. Moreover the results may vary each time the algorithm is run. This will be further explained in Chapter 4.

2.6 Temporal Network Analysis in Longitudinal Data

When data are available as a date-annotated series, it is possible to conduct a time-based analysis. As introduced in Section 1, this may provide relevant insights into the underlying dynamics of collaboration over time, through the evolution of centrality measures and communities.

In such cases, the data can be divided into discrete time intervals (e.g., years) to create a set of networks $G = \{G_1, \dots, G_t, \dots, G_n\}$, where each network G_t represents collaborations or interactions within a specific time interval. The generic network for any year is denoted as G_t , or G_y , with the subscript y indicating the corresponding year.

Within each yearly network, centrality measures and partitions can be calculated, and our primary objective is to understand how they evolve over time. With reference to node-properties such as strength degree or coreness, we can compare the trajectory of each node along time.

For communities, the method involves comparing each community $C_{i,y}$ with each $C_{j,y+1}$, and determine whether C_i and C_j are disjoint or have a non-null intersection. If they intersect, the relationship between the two communities is further classified as either continuing (i.e. C_i shares most of its members with C_j), or as part of a merge or split. To ensure accurate tracking across different years, we assign global community labels that remain consistent for identical or continuing communities.

This method enables the tracking of network properties over time, allowing for the analysis of evolving patterns in relationships, such as the growth or decline of collaborations between organizations.

However, it is important to note that nodes may not appear in every year. Conse-

quently, the size of the network—measured by the number of nodes (n_v) and edges (n_e)—can vary across time periods, and this fluctuation must be analysed carefully to account for changes in network composition and structure.

2.7 Community detection algorithms

Community detection is a crucial step in network analysis, as it helps to understand the role of an organisation within the network.

A community detection algorithm $\mathcal{A}(G, \rho) \rightarrow \mathcal{P}$ is a function that takes as input a graph G and one or more parameters ρ , and returns a partition \mathcal{P} .

Many methods exist to detect meaningful community structures based on the density of their interconnections [42, 23, 48]. This principle is quite general although other attachment and aggregation mechanisms are possible in social networks [54]. The main strategies for identifying an optimal partition include the detection of actors or edges with high centrality [63] optimization-based algorithms [22], statistical inference using stochastic block models [52], dynamic process-based approaches such as random walks [84]. Furthermore, a new class of community detection methods has emerged that exploits node semantics or node attributes in addition to network topology. According to the taxonomy proposed by [47], these include graphical model-based community detection, deep learning-based community detection, as well as node embeddings [88].

Although many of these methods focus on partitioning networks into non-overlapping communities, there is a diverse range of variants, including hierarchical clustering [18] [71], which captures structures at different scales, overlapping communities [67] [4] [73] and mixed-membership communities [1], where a node can belong to more than one community, as well as a combination of overlapping and non-overlapping communities [56]. However, probably due to their ability to produce easily interpretable results, optimisation methods that generate non-overlapping partitions are still widely used.

Research has investigated detectability thresholds, resolution limits (which limit the ability to find small communities in large networks), the generation of disconnected communities [90], and the computation time and cost on large networks.

While many more algorithms are discussed in the literature, their source code is not always openly available, limiting their practical application. In this section, we focus on the community detection algorithms available in the `igraph` library [21], that produce non-overlapping partitions: Infomap (IM), Leiden (LD), Louvain (LV), Label Propagation (LP) and Walktrap (WT). These algorithms have been tested and discussed by literature, as for example in [39], and [76].

LV The Louvain (LV) algorithm [9] optimizes modularity using a greedy approach.

Initially, each node is assigned to a separate community; nodes are then iteratively moved to the community of one of their neighbours, maximizing the positive impact on modularity, until no further improvement can be made. LV yields stochastic results, as it relies on random initialization to determine the sequence in which nodes are examined, and identifies a local maximum of modularity. The algorithm has one parameter, called resolution (r) that controls the size of detected communities: $r > 1$ leads to smaller and more numerous communities, while $r < 1$ leads to larger and fewer communities. The LV algorithm can take into account edge weights, but is compatible only with undirected networks.

LD The Leiden (LD) algorithm, as introduced in Traag et al.'s work [90], is a community detection algorithm primarily designed as an enhancement of the Louvain method, to mitigate the generation of disconnected communities. Notably, it shares similarities with the LV algorithm, employing a resolution parameter and yielding stochastic results. Similarly to the previous algorithm, also LD can be applied only to undirected networks.

IM The Infomap algorithm [83, 82, 25] exploits the information-theoretic duality between finding community structure in networks and minimizing the description length of a random walker's movements on a network; communities are aggregated following an approach similar to LV, using a new random sequential order at each iteration, hence results are stochastic.

WT Walktrap [74] is a hierarchical clustering algorithm based on the assumption that nodes within a community are likely to be connected by shorter random walks. Beginning with a non-clustered partition, it merges adjacent communities minimizing the squared distances between each node and its community, iterating until no further improvement is possible. A user-defined parameter s defines the length of the random walk to be performed, controlling the resulting community size.

LP Label Propagation (LP) relies on the notion of proximity or neighborhood relationships, as discussed in [79]. Initially, each node is assigned a unique community label, then nodes are iterated through in a random sequential order, and each node adopts the label that is most prevalent among its neighbors. This process continues until each node shares the label of the majority of its neighbors.

The Louvain algorithm is widely employed for community detection in networks due to its efficiency, scalability, and the intuitive nature of its greedy method, making it worthy of further examination. Since its introduction, numerous refinements

have been proposed in the literature to address its limitations; three key studies are presented to illustrate these advancements. The retrospective study *“Fast Unfolding of Communities in Large Networks: 15 Years Later”* [8] highlights the algorithm’s limitations (namely the resolution limit, sensitivity to initialization, and susceptibility to local optima) and presents the main optimizations proposed, which include generalizations of quality functions and parallelization techniques. The paper also emphasizes the role of vertex processing order, noting:

“In most implementations of Louvain, the order in which vertices are considered is random. Although this can be seen as a problem, it does allow us to explore more possible solutions. If the aim is to obtain an identical partition, any fixed order can be used.”

The second study, *“An Improvement on the Louvain Algorithm Using Random Walks”* [24] introduces the Random Walk Graph Partition Louvain Algorithm (RWGP-Louvain). This method enhances modularity optimization by adding a random walk phase, achieving higher modularity on graphs with ambiguous structures. However, variability and solution multiplicity remain an issue with RWGP-Louvain, as inherent to all modularity-based approaches. The third study, *“An Improved Louvain Algorithm Based on Node Importance for Community Detection”* [2] proposes the Improved Louvain Algorithm (ILVA). By replacing random vertex processing with a deterministic order based on degree centrality, ILVA stabilizes results, producing consistent community structures and higher modularity values across runs. It must be noted that this approach sidesteps randomness rather than addressing its underlying impact.

The issues related to variability and node ordering will be discussed further in Chapter 3, and will be the basis for introducing a new framework for community detection that explicitly examines and addresses the role of vertex ordering in shaping results in Chapter 4.

2.8 Benchmark networks

Benchmark networks are synthetic graphs with known properties designed to test and validate algorithms in network analysis. The use of benchmark networks is crucial because they provide a standardized framework for assessing the effectiveness of algorithms in tasks such as community detection, and assessment of centrality indicators.

While benchmark networks usually have a predefined "ground truth" to measure algorithm accuracy, this thesis assumes no such ground truth exists. Instead, the focus is on evaluating algorithms based on their ability to reveal meaningful patterns

and insights, emphasizing performance and interpretability without relying on a predefined standard.

Three networks will be used for test and performance evaluation: Zachary's **Karate** club network [93], (**LFR**) and Ring of Cliques (**RC**).

The **Karate** network is a real-world example, though small in size, it is not trivial. It has become a standard test case in the field of community detection due to its well-documented structure and the interpretability of its communities. Despite its simplicity, the Karate network provides valuable insights and is frequently used as a reference point for evaluating the performance of algorithms across different studies.

In contrast, the **LFR** and **RC** networks are artificial benchmark networks designed specifically to test and compare the performance of community detection algorithms.

LFR LFR is a parametric benchmark network, named after the Authors that first proposed it Lancichinetti – Fortunato – Radicchi [51]. It is widely used as benchmark for testing the performance of community detection algorithms as it is characterised by a power-law distribution of the degree of the nodes (parameter τ_1) and the size of the communities (parameter τ_2). For the purpose of this thesis the LFR benchmarks will be used with parameters $N = 1000$ nodes, $\tau_1 = 2$, $\tau_2 = 3$, an average degree = 10, community size between 20 and 50, and nominal mixing parameter in the range $\mu \in (0.05, 0.50)$. Lower values of mixing parameter μ indicate that the communities are sharply separated and are therefore easily identified by community detection algorithms; on the contrary, high values of μ are related to networks with fuzzy communities that are hard to identify.

RC The Ring of Cliques is another artificial model that offers a simplified, yet controlled environment to study algorithm performance. RC is a benchmark network composed of k_0 identical cliques of size s , where pairs of cliques are connected in a regular sequence to form a ring. A family of RCs, with a fixed s and varying k_0 , provides a valuable benchmark for community detection as it ensures a consistent degree of fuzziness with a mixing parameter $\mu = 1/s!$. A RC is apparently a straightforward problem for community detection algorithms, which can be expected to identify each clique as a community. However, it can become a more challenging problem when additional nodes are introduced such as 'bridge nodes' between pairs of cliques or a central node connected to each clique. Such additional nodes will result in a slight increase in μ (while keeping it independent of k_0), and create a dilemma for the community detection algorithm since bridge nodes are equally connected

to two communities and central nodes are symmetrically connected to each clique.

It is important to note that while benchmark networks are a convenient tool for testing and explaining the behaviour of algorithms, real-world networks are often far more complex and present unique challenges that are not captured by these simplified models. As a result, benchmark networks are ideal for initial experimentation and validation, but they may not fully represent the intricacies of real-world systems.

Limitations of community detection

This chapter examines the limitations of community detection, addressing both well-known challenges, such as the variability observed across repeated trials (section 3.1), and less-explored issues, including the validity of the results (3.2), outliers (section 3.3, and the influence of input ordering on the outcomes (3.4).

3.1 Variability

In networks with simple topologies, community detection algorithms generally produce consistent results. However, in networks with fuzzy or complex community structures, significant variability can occur both across different algorithms and within repeated runs of the same algorithm.

Variability is a critical issue that compromises the reliability of conclusions drawn from community detection analyses, while also hindering the replicability and verification of results.

The first cause of variability can be explained by the fact that each algorithm relies on different principles and assumptions about what defines a community. This issue can be mitigated by standardizing the analysis through the selection of a single algorithm.

However, the second cause of variability is more complex and occurs specifically in algorithms that rely on heuristic or randomized methods to explore only a subset of all possible solutions. Formally, this is expressed by the fact that the algorithm $\mathcal{A}(G, \rho)$ can produce different partitions $\mathcal{P}_i \neq \mathcal{P}_j$, even when using exactly the same set of parameters ρ is used.

Figure 3.1 illustrates the variability of results obtained by different algorithms (LV, LD, IM, WT and LP) on a LFR benchmark network characterised by a nominal value of mixing parameter $\mu = 0.40$. Partitions and number of communities are

different at each trial, and modularity is not sufficient to identify a single optimal solution.

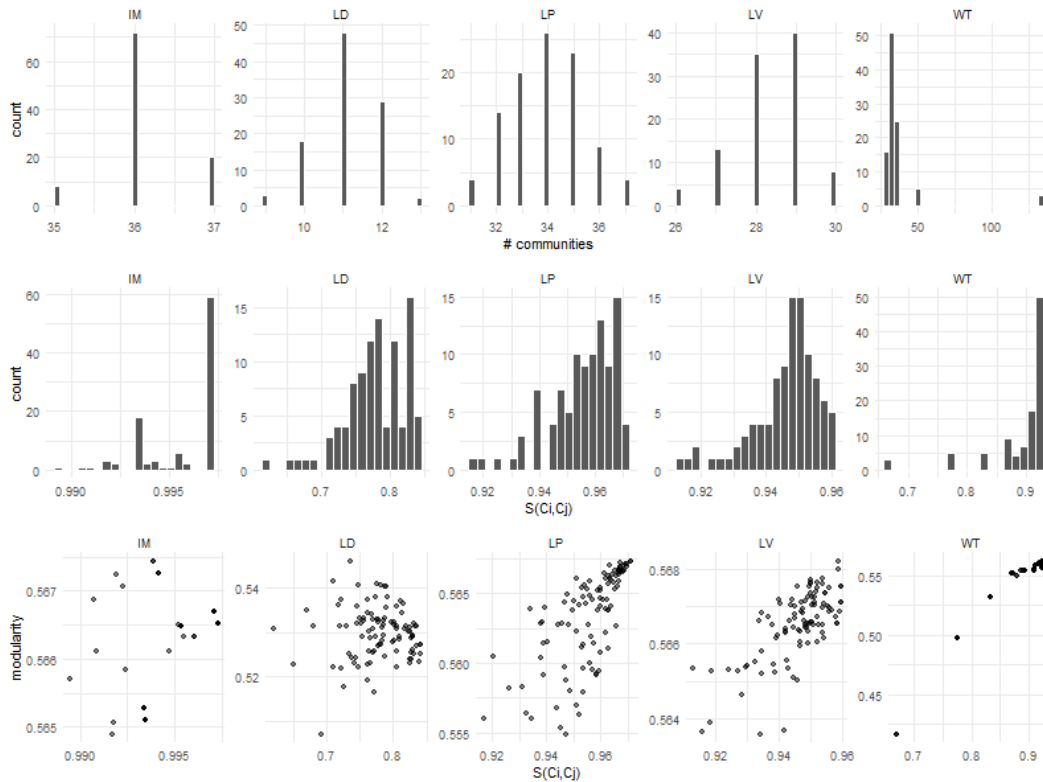


Figure 3.1: Variability of results of selected community detection algorithms on a LFR benchmark network with a nominal mixing parameter $\mu = 0.40$. Top: distribution of the number of communities. Middle: similarity between pairs of partitions. Bottom: scatterplot modularity and similarity.

These results suggest that relying on a single execution of an algorithm may be insufficient even for simple tasks such as determining the number of communities or assessing whether a given pair of nodes belong to the same community.

The relevance of the issue, however, depends on the specific goals of the analysis. If the focus is on evaluating the overall performance of the algorithm, as for example in Stoltenberg et al. [89], a common solution is to run the algorithm multiple times (e.g., $t = 100$) and compute performance metrics based on the mean and standard deviation of the results.

On the other hand, if the objective is to answer specific questions like "Do nodes u and v belong to the same community?", variability becomes a more significant challenge. The answer may change with each run of the algorithm, leading to inconsistent and non-reproducible results.

However, variability should not be seen as a flaw in the algorithm. Instead, it can provide valuable insights into the network's underlying structure. As Fortunato and Hric [41] discuss, variability across repeated trials can be leveraged to improve results when combined with consensus methods.

Ensemble and **Consensus** approaches have been proposed based on the idea that combining results from multiple partitions can improve the stability and reliability of the outcomes. This will be further discussed in Section 4.2.

3.2 Validity of results

Community detection algorithms always produce a partition of the network, assigning each node a membership label linking it to a specific community. However, they do not provide any feedback on the quality of the result. For the purposes of this thesis, we will validate the resulting partition by checking the following criteria:

- a. A community should consist of more than one node.
- b. A community should be smaller than the entire network.
- c. A community should be internally connected
- d. A community should be composed of nodes that are more densely connected with other nodes within their own community than with nodes outside of it.

Further discussion of these issues, including challenges related to exploring the solution space and the importance of quality checks for validating the results, will be presented in the sections on Solution Space (Section 4.1) and Quality Check Functions (Chapter 5).

Criteria a) and b) suggest that \mathcal{P} should have a meaningful number of partitions. In most real-world networks, encountering one or a few **singletons** can be a normal outcome, especially if those singletons are isolated components of the network. Moreover, singletons may be the expected output when they have a special role within the network, acting as bridges between different communities, as the central node in Figure 4.8 discussed in Section 4.1.

Nevertheless, some extreme scenarios can arise in the application of community detection algorithms, both representing degenerate cases where the detection of a community structure is not possible, and the partition fails on the first and second quality criteria:

$k = n_v$ This scenario occurs when each node is placed in its own community, meaning k , the number of communities, equals n_v , the total number of nodes in

the network. Essentially, every node is treated as a singleton, isolated from all others. While having one or more singletons is not unusual if these nodes are disconnected components of the network, this extreme case of "all singletons" indicates that the algorithm failed to detect any cohesive groupings within the network.

$k = 1$ At the other extreme, the algorithm may return a single community, assigning all nodes to the same label. This can happen in random or highly interconnected networks where no distinct clusters or subgroups are present.

In the case of random networks, which lack an inherent community structure by definition, the most appropriate outcome would be a degenerate partition and indeed algorithms such as IM and LP to random networks consistently yields $k = 1$. However, other algorithms, including LV, LD and WT applied to random networks may generate partitions with $k > 1$, erroneously suggesting the presence of multiple communities.

Criterion c) requires that every community must be internally connected (i.e., each node should be reachable from a given other nodes in the community). While this might seem straightforward, it is not always upheld in practice. For example, the Louvain (LV) algorithm, which optimizes modularity, can assign distant, unconnected nodes to the same community because it prioritizes global structural patterns over local connectivity [90]. This can lead to communities consisting of multiple disjoint subgraphs. Similarly, the Leiden (LD) algorithm and the Label Propagation (LP) algorithm can also produce internally disconnected communities, as discussed by Sahu et al. [85] and highlighted in some of the examples in Chapter 7.

The issue of disconnected communities is particularly problematic because software packages such as the `igraph` library do not check for or flag such cases. As a result, a partition that includes communities failing to meet the connectivity criterion without any indication that the result is invalid. Therefore, it is crucial to perform a post-algorithm check to ensure the validity of the generated communities.

Criterion d) can be assessed using the mixing parameter μ that indicates the extent to which nodes within a community are connected to nodes outside that community. As described in section 2.3, $\mu > 0.5$ implies that on average the nodes have more external than internal connections, which violates the fourth validity criterion. Hence, partitions with $\mu > 0.5$ should be flagged and discarded.

Additionally, while μ is typically calculated as an average for the entire network, a more granular analysis could involve computing μ for each community individually. This would allow for a more detailed assessment of whether certain communities are poorly defined, even when the overall network's mixing parameter appears

reasonable.

3.3 Outliers

An additional focus is to examine the behaviour of community detection algorithms when confronted with *outliers*, which are nodes that display significantly different behaviour compared to the rest of the network.

Outliers can be highly relevant for interpreting the community structure, for example in a social network it is the case of an individual that is well-connected to many actors that belong to different communities. Another example can be found in networks modelling organizations involved in knowledge flows for innovation. Here, an organization that acts as a bridge between two communities—such as a consultant or a key service provider—can be considered an outlier, playing a crucial role in fostering innovation by connecting otherwise disconnected groups.

Not all algorithms have the capability to detect outliers. For example, algorithms based on modularity maximization consistently form communities with two or more nodes and suffer from the resolution limit. This issue is well-documented in the literature. Fortunato and Barthélemy [40] explain how modularity optimization tends to overlook smaller communities, leading to the merging of outliers into larger groups. Other algorithms have the opposite behaviour and identify outliers as singletons.

To categorize these varied approaches, a novel taxonomy is introduced in this thesis, classifying algorithmic responses to outliers into three categories, illustrates in figure 3.2:

- **Incorporate:** In this approach, the outlier is forced to be part of a larger community, keeping the number of communities k as low as possible.
- **Highlight:** in this approach, outliers are identified and labelled as singleton. This allows to analyse their distinctive role, but has the drawback of generating a high number of communities.
- **Group:** This involves the identification of individual outliers and label each of them with a conventional label l_0 . This allows to identify clearly all outliers, and at the same time has little impact on k . It must be noted that the group of outliers is internally disconnected, hence is not a proper community.

in Figure 3.2, a RC with $k_0 = 4$, $s_0 = 6$, bridge nodes, and a central node provides a clear representation of the three alternative ways to manage outliers, This example shows clearly how each of the strategies can support the analysis. *Incorporating* outliers correctly detects the number of communities ($k = k_0$), but overestimates s and does not capture symmetry. On the other hand, *highlighting*

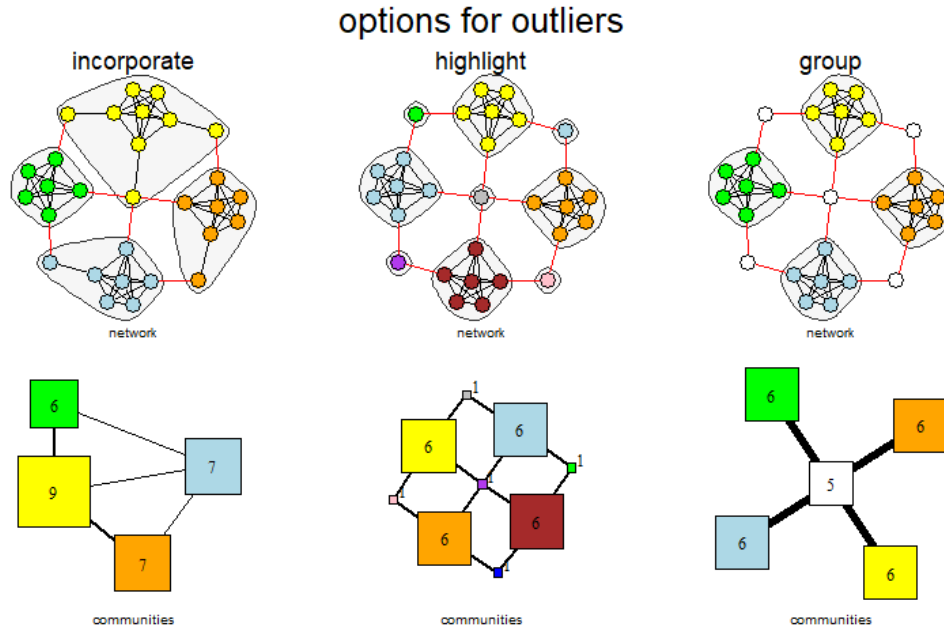


Figure 3.2: Three alternative strategies to manage outliers: incorporate (left), highlight as single-node communities (centre), or group into an outliers' community (right). The top row shows the network; the bottom row shows a graph of the communities, labelled with the number of nodes in each community.

perfectly recognizes community size $s = s_0$, but at the cost of overestimating their number ($k = (2k_0) + 1$). Finally, *grouping* provides a trade-off between the previous options, capturing community size and symmetry while adding only a fixed bias to their number ($k = k_0 + 1$).

3.4 Input ordering bias

A less known issue is the **input-ordering bias**. Although networks are mathematically non-ordered entities, their implementation in any software algorithm is inevitably ordered (in the form of a list of edges, or a matrix). The order in which nodes and edges are stored in the practical implementation of the network can affect the results, as illustrated in [62].

Ideally, community detection algorithms should ignore order, but this is not always the case in practice. The issue can be highlighted by comparing $\mathcal{P} = \mathcal{A}(G)$ with $\mathcal{P}^* = \mathcal{A}(G^*)$, where G^* is generated by a random permutation of edges and vertices of G . If \mathcal{A} is unbiased algorithm, we may expect that $\mathcal{P} = \mathcal{P}^*$. In complex, real-world networks the differences \mathcal{P} and \mathcal{P}^* may not be noticed.

The input ordering bias can be devised using the following test: let G be a network

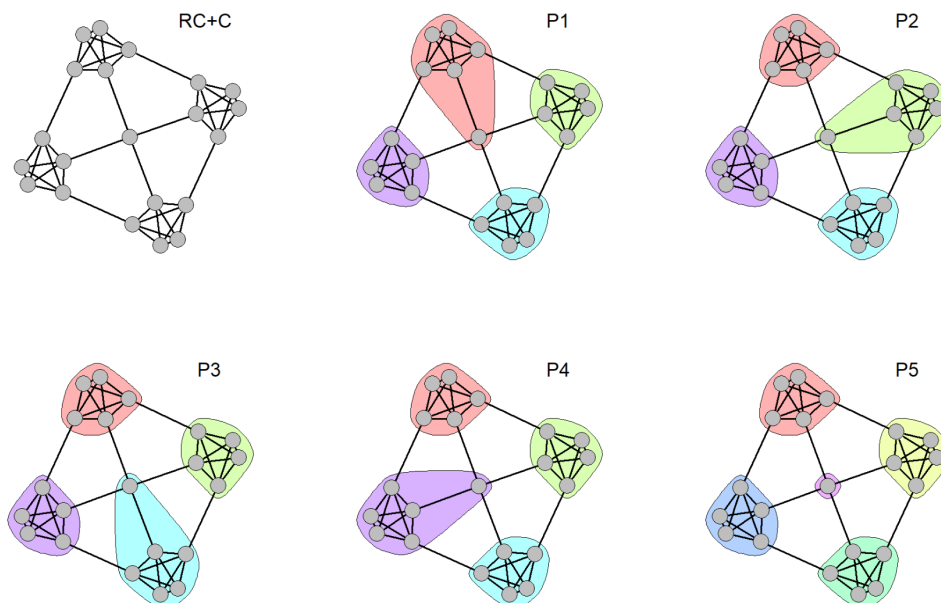


Figure 3.3: Ring of cliques with central outlier (RCC) and resulting partitions. The RCC network comprises four cliques and a central outlier node connected to all cliques. Partitions \mathcal{P}_1 through \mathcal{P}_4 assign the outlier to one clique, introducing imbalance. Partition \mathcal{P}_5 isolates the outlier as a single-node community, a configuration not all algorithms can produce.

with nodes and edges built in known order, with a sharply defined and symmetric community structure, and identifiable outliers, such as the RC with $nc = 4$ and $cs = 5$, depicted in Fig 3.3. The central node is connected with equal strength to four communities, hence one would expect that an unbiased algorithm assigns it to any of the four communities with equal likelihood.

Most algorithms are likely to generate partitions such as \mathcal{P}_1 , \mathcal{P}_2 , \mathcal{P}_3 , or \mathcal{P}_4 where the central node (outlier) is incorporated into one of the cliques. However, these solutions are inherently flawed, as they unfairly favour one clique over the others. Alternatively, a solution like \mathcal{P}_5 , where the outlier forms a separate community, may appear more equitable. Nevertheless, not all algorithms are capable of producing such a partition, as it requires accepting the assumption that a single node can constitute a valid community.

Fig 3.4 shows the results of a test with different algorithms (IM, LD, LV, LP, LV, WT), over 100 iterations. We observe that most algorithms exhibit a noticeable input-ordering bias, with the exception of LP. Specifically, when applied to G , IM assigns the centre to a single-node community, WT always to the same community C_2 , while LD and LV strongly favour community C_1 . In contrast, when applied

to G^* , all algorithms produce a less biased scenario. Specifically, LV, WT, and LP consistently integrate the outlier node into an existing community and generate partitions P1 to P4 with equal likelihood, depending on the random seed. On the other hand, IM and LD are capable of highlighting the outlier: they not only generate P1 to P4 with equal likelihood but can also produce P5, although the latter occurs with a lower probability.

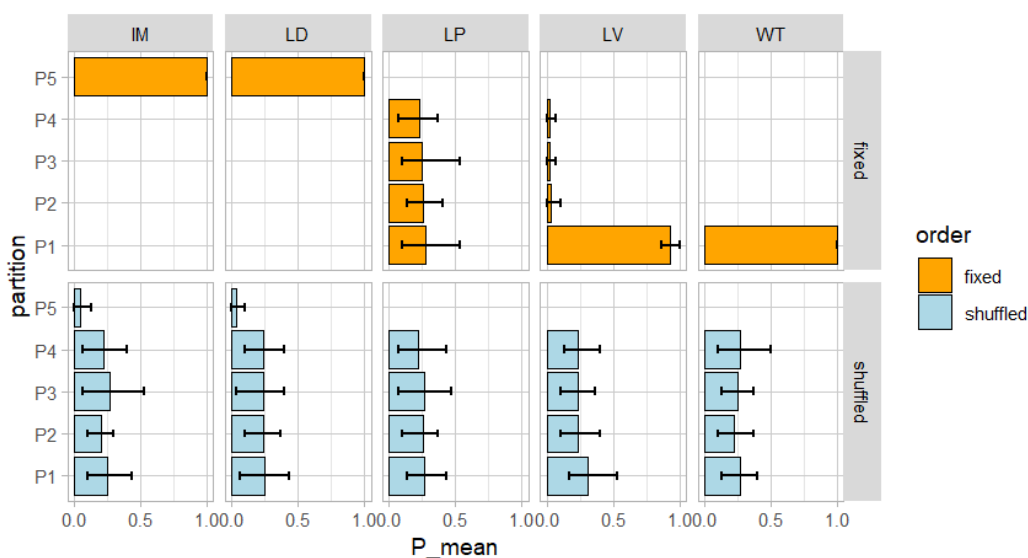


Figure 3.4: An illustration of input-ordering bias, using a RC with $nc = 4$, $cs = 5$ with a central node. Above: label assigned to the central node by various algorithms, applied $t = 100$ times to network G . Below: labels assigned to the central node applied to network G^* , a copy of G randomly permuted at each iteration. Labels: S = the centre is highlighted as a single-node community, C_i = the centre is incorporated in community i .

Input ordering bias has been discussed in the literature, notably by [14, 44, 15] mainly focusing on modularity-based methods. In this paper, we aim to generalize these results to any algorithm, and to devise a procedure that mitigates input-ordering bias, while improving the stability and reliability of results.

Enhancing stability of community detection

This chapter is taken from the papers "Enhancing Stability and Assessing Uncertainty in Community Detection through a Consensus-based Approach" [62], currently under peer review, and "Beyond One Solution: The Case for a Comprehensive Exploration of Solution Space in Community Detection" [61], accepted for the Complex Networks and their Applications conference - December 2024.

The preceding chapters have addressed the significance of networks as models for collaboration, discussed fundamental concepts in network analysis, and highlighted several limitations. This chapter presents a novel workflow that aims to overcome the limitations presented in previous chapter, and enhance the reliability and replicability of community detection. The method is built on the following principles:

- **Solution Space Exploration:** Each execution of $\mathcal{A}(G, \rho)$ identifies a point within a solution space. if G is simple and has a clear community structure, this may represent a single, definitive solution. However, in more complex scenarios, the solution space may be more intricate and require thorough exploration. A taxonomy for the solution space is introduced in 4.1.
- **Consensus:** When multiple solutions emerge, a consensus approach becomes necessary to synthesize the various possibilities into a unified solution. This is introduced in Section 4.2.
- **Uncertainty** is inherent in community detection; therefore, the community structure should be represented by a list of triplets (v, l, γ) , where uncertainty is explicitly acknowledged.

The proposed workflow is illustrated in Figure 4.1.

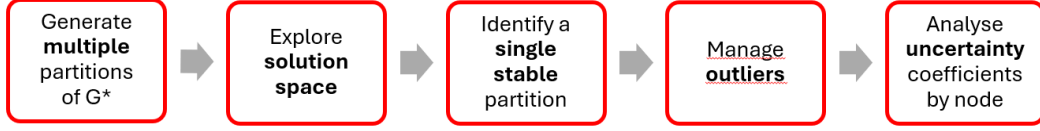


Figure 4.1: proposed workflow to identify a stable and valid partition of a network.

4.1 Solution Space

This section explores the importance of examining the solution space in community detection, highlighting its role in achieving reliable results when dealing with real-world problems.

As discussed in Section 3.1, a community detection algorithm $\mathcal{A}(G, \rho) \rightarrow \mathcal{P}$ is a function that takes as input a graph G and one or more parameters ρ , and returns a partition \mathcal{P} . Ideally, \mathcal{A} should produce a single, valid partition each time it is applied with the same parameters. In practice, however, for large, dense networks, \mathcal{A} may produce different partitions, $\mathcal{P}_i \neq \mathcal{P}_j$ at each trial.

The **solution space** $\mathbb{S} = \{\mathcal{P}_1, \dots, \mathcal{P}_{ns}\}$ is the set of all unique partitions that \mathcal{A} produces across t trials.

The aim of exploring solution space is to determine the minimum number of trials t_c required to confidently assert that \mathbb{S} is *stable*, i.e. it is unlikely to expand with an additional run of \mathcal{A} .

4.1.a Bayesian model

To understand whether \mathbb{S} is stable and the relative frequency of each of the solutions found, we created an experimental setting, described in Algorithm 1, to produce a probability model \mathbb{M} under a Bayesian framework. The algorithm implements a Beta-Binomial model \mathbb{M} within a Bayesian framework. The solution space is explored by successive trials, and each trial is treated as a Bernoulli process, with the Beta distribution updating the probability of finding new solutions. As trials progress, the model refines the estimate of p_{stable} , optimizing computational resources and allowing the experiment to stop when further trials are unlikely to reveal new solutions.

The model \mathbb{M} is composed of a probability distribution θ_i associated to each partition $P_i \in \mathbb{S}$. The process starts with one solution, and therefore, only one probability distribution θ_1 ; as new solutions are found, additional distributions are added to the model, and the process continues until either the solution space has reached stability, or a maximum number of trials is reached.

Each trial is modeled as a Bernoulli experiment with two possible outcomes:

At each trial, \mathbb{M} is updated using the observed successes and failures: assuming the k -th solution is a success, the corresponding posterior distribution is updated incrementing the success count $\alpha_k \leftarrow \alpha_k + 1$ while all the others are updated incrementing the failure count $\beta \leftarrow \beta + 1$.

From each $\theta_i \in \mathbb{M}$ we can derive a confidence interval $[p_{i,lower}, p_{i,upper}]$ and a point estimates \bar{p}_i of the likelihood of any solution P_i . Specifically:

$$\bar{p}_i = \mathbb{E}(\mathbf{Beta}(\alpha, \beta)) = \frac{\alpha}{\alpha + \beta}$$

where \mathbb{E} is the expected mean of distribution.

The probability that \mathbb{S} is stable after t trials (meaning the $t + 1$ trial will not yield a new solution) can be modeled as $p_{stable} = 1 - \mathbb{E}(\beta(t + 2, t - ns + 2))$.

The exploration of solution space continues until either t_{max} is reached or the p_{stable} reaches a predefined threshold τ .

An important step in the exploration of \mathbb{S} is highlighted in step 4 of the algorithm: G should be permuted at each trial, to avoid incurring in the *input ordering bias*, as discussed in [62]. Moreover, a partition P may be considered **invalid** for several reasons: it may be trivial (e.g., $k = 1$ or $k = n_v$), internally disconnected, or fail to meet the community definition (i.e. nodes in C_i are more connected to nodes in any other partitions C_j (where $j \neq i$) than within C_i).

4.1.b Taxonomy

Finally, a taxonomy can be introduced to classify \mathbb{S} in different categories (depicted in Figure 4.2), based on ns and the relative frequencies of P_i observed in the trials.

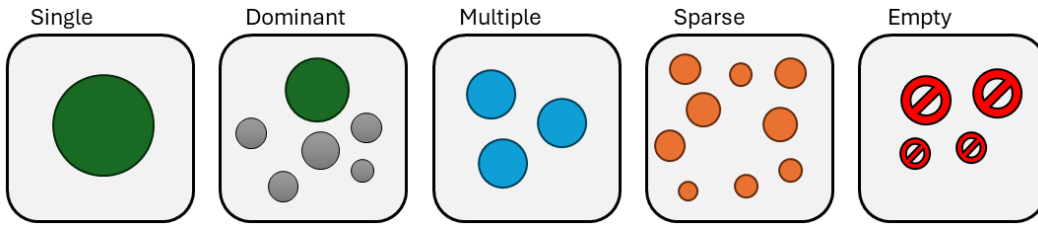


Figure 4.2: Taxonomy for the solution space of generated by a community detection algorithm.

The taxonomy proposed in Figure 4.2 can be formally defined as follows: the *Single* category describes the case where the solution space is stable and there is only one valid partition ($ns = 1$). The *Dominant* category occurs when there are multiple valid partitions ($ns > 1$), but one partition is dominant, i.e., $\max(p_{lower}) > 0.5$. The *Multiple* category applies when $ns > 1$ and $\max(p_{lower}) < 0.5$. The *Sparse* category

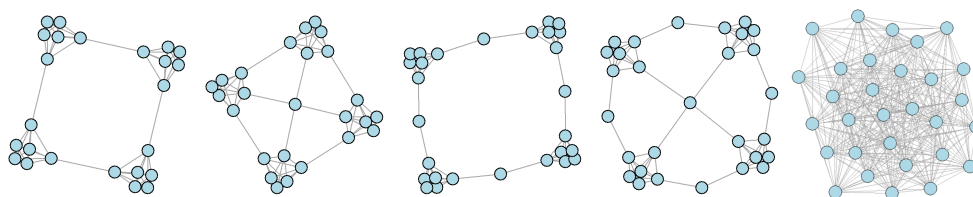


Figure 4.3: Sample networks used in the experiment. Left: A simple ring of cliques. Middle: The same ring of cliques with an added central node. Right: A variation of the original ring of cliques with added bridge nodes. While the number of true communities remains 4 in all cases, the additional nodes significantly increase the complexity for the community detection algorithm.

occurs when a high number of solutions exists ($n_s \approx t$) each with a low probability ($\max(p_{upper}) \approx 0$). Lastly, the *Empty* category represents a situation where there are no solutions, or all solutions are invalid, i.e $n_s = 0$.

4.1.c Examples

To illustrate the diversity in the structure of solution spaces, we employ a set of artificial networks referred to as Rings of Cliques (RC). These networks are particularly useful in studying the effects of different topological changes on community detection and network analysis.

The first network represents a very simple case, where the solution is intuitive with $k = 4$ communities, making it easy for any algorithm to identify the structure. However, in the subsequent examples, the addition of a central node or bridge nodes introduces greater complexity, making the community boundaries less clear. Finally, we include a random graph, where no community structure is present, to illustrate the case of an Empty solution space.

Single

In this example, we consider a simple model of a ring of cliques: 4 cliques, each containing 6 nodes. This basic configuration is used to illustrate the process of exploring the solution space of community detection algorithms. The threshold for stability is set at $\tau = 0.95$, and the maximum number of trials is $t_{\max} = 200$. The solution space stabilizes at $t = 50$, meaning that after 50 trials, it is unlikely that additional trials will yield new solutions or alter the current understanding of the solution space.

A Bayesian model is applied at each trial, updating the likelihood that the solution space is stable. As shown in the figure, the ribbon represents the narrowing confidence intervals, reflecting that with more trials, our certainty about the solution space increases.

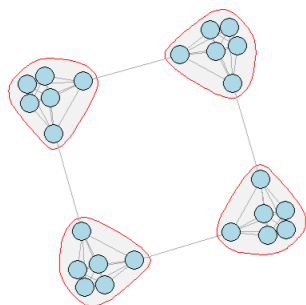


Figure 4.4: Example of a Single solution space. The network consists of 4 cliques, each containing 6 nodes. Any community detection algorithm consistently identifies the same solution, yielding 4 distinct communities. This illustrates a stable solution space, where no variation in community structure is observed across trials.

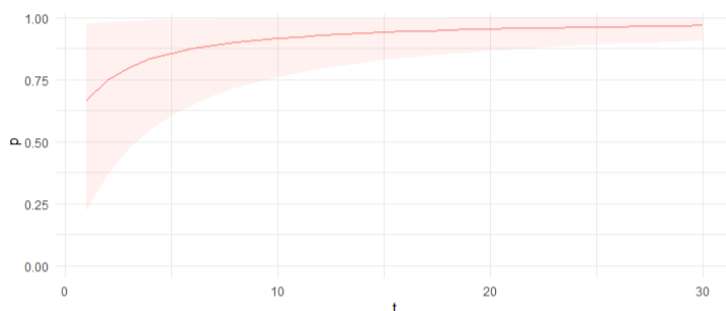


Figure 4.5: Evolution of the knowledge on solution space. Parameters: $t_{max} = 200, \tau = 0.95$. Result: \mathbb{S} is stable at $t = 50$ trials, after which no new solutions emerge. The Bayesian model is applied at each step, and the narrowing ribbon represents increased confidence as the solution space stabilizes.

Dominant

The "dominant" solution space is characterized by the fact that multiple independent solutions are found, but one particular solution is generated far more frequently than the others. This occurs when a solution P_i appears in a majority of trials, with its estimated probability $p_i > 0.5$.

Figure 4.6 shows an example, referring to RC_c : Solution 1 is by far the dominant one (93%), but there are two alternatives with $p_i = 0.3$ and 0.5 respectively.

Figure 4.7 shows how the knowledge about solution space evolves as trials progress. The parameters are $t_{max} = 200, \tau = 0.95$, and the search actually stops at $t = 80$. It may be tempting to select Solution 1 as the "best" solution, given its dominant frequency, and to treat it as the only valid partition. However, there are two important issues to consider.

The first issue is that a more frequently found solution does not necessarily mean it is the "best" or most optimal partition. The concept of an optimal solution is not

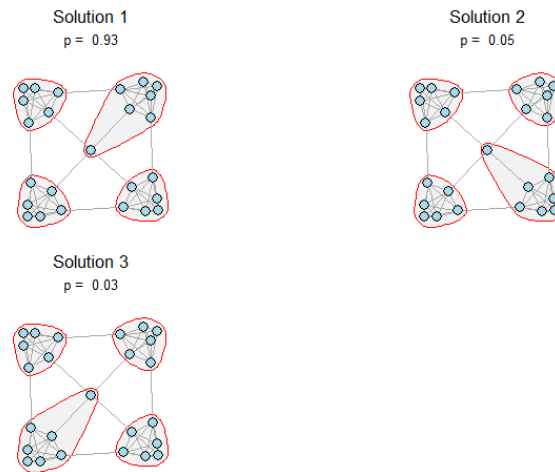


Figure 4.6: Solutions present in \mathcal{S}

intrinsic to the solution space itself but instead depends on the specific analysis framework. For instance, the "best" solution might be the one with the highest modularity score, or it may be defined by another objective function relevant to the network under study. In other words, the determination of the "best" solution is shaped by external criteria reflecting the goals of the analysis, not merely the solution's frequency.

It is possible, and not uncommon, for a community detection algorithm to frequently return a suboptimal solution due to its heuristic or randomized nature. The optimal partition—according to some metric—might only rarely emerge in the trials. This highlights the importance of not conflating solution frequency with quality: a dominant solution, while stable and recurring, may not represent the most accurate or desirable partition for the problem at hand.

Figure 4.7 illustrates this concept, showing how one solution dominates in frequency but does not inherently indicate that it is the best according to any specific criterion. The Bayesian framework aids in quantifying the dominance, but the decision about which solution is optimal remains a separate analytical judgment.

The second issue is the input ordering bias. As shown in Figure 4.7, there is a clear symmetry in the network, but the solutions do not reflect this symmetry. One solution is favored over others with no apparent reason. This bias arises because the algorithm is sensitive to the ordering of nodes and edges in the software object used. However, the mathematical definition of the network treats it as an unordered entity. The algorithm's sensitivity to input order is an undue bias, as it relies on the inherent ordering of data structures (like vectors) used in the software implementation.

The input ordering bias can be addressed by randomly permuting the order of

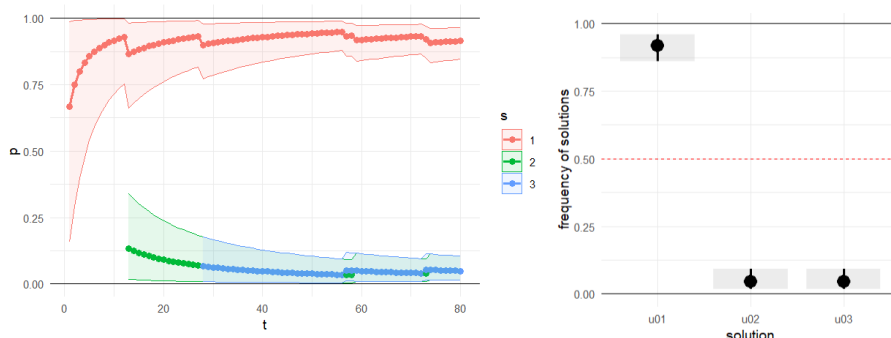


Figure 4.7: Evolution of the knowledge on solution space. Parameters: $t_{max} = 200, \tau = 0.95$. Result: \mathbb{S} is stable for $t = 80$, with 3 solutions.

edges and nodes within the network at each trial. This technique is applied in the next example to eliminate the bias and ensure a more accurate representation of the solution space.

Multiple

When the same RC_c network is analyzed using the same community detection algorithm, but with random permutations applied at each trial, the solution space changes dramatically. In this case, the algorithm identifies four distinct solutions, which respect the inherent symmetry of the network. These solutions are equally likely, with each having a mean probability of approximately 0.25. Specifically, the Bayesian interval estimates for each solution overlap, indicating no clear dominance. Figure 4.8 illustrates this balanced solution space.

In this experiment, we set $t_{max} = 200, \tau = 0.95$, with the stability criterion being met around $t = 100$. The evolution of the solution space as a function of trials is different from the previous case and is shown in Figure 4.8. Here, the algorithm gradually converges on the four equally likely solutions, which more faithfully represent the community structure.

This more balanced outcome reflects the symmetry present in the network, where no solution is preferred over the others. The central node, which connects the otherwise symmetrical substructures, cannot be preferentially assigned to any specific community. It is, therefore, more appropriately treated as an outlier or a singleton community. This highlights that the small differences in modularity between these solutions do not capture the true nature of the solution space.

In cases like this, a consensus-based approach to community detection might be more suitable. For example, the central node could be consistently identified as its own singleton community, rather than arbitrarily included in one of the symmetrical subgroups. Such an approach would better represent the underlying structure of

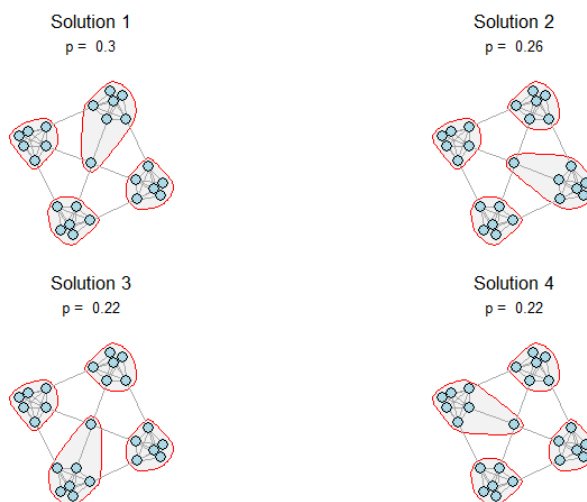


Figure 4.8: Solutions present in \mathbb{S}

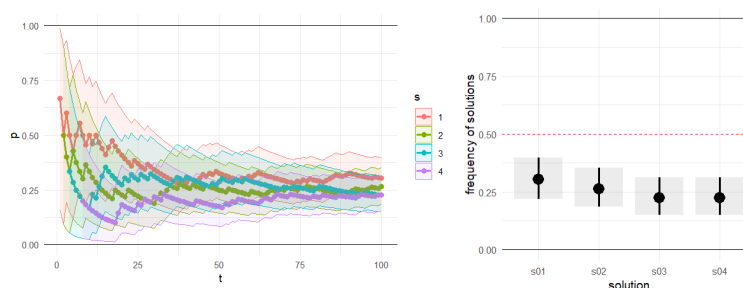


Figure 4.9: Evolution of the knowledge on solution space. Parameters: $t_{max} = 200$, $\tau = 0.95$. Result: \mathbb{S} is stable at $t = 100$. The credible intervals are overlapping, hence all four solutions are equally likely.

the network, as it avoids forcing the central node into a community where it does not naturally belong.

This example demonstrates how permuting the network at each trial can lead to a more accurate and faithful representation of its community structure. In contrast to the "dominant" solution space, the "multiple" solution space provides a more nuanced view, revealing the potential symmetries and allowing for more equitable consideration of different partitions. It also underscores that relying solely on modularity as a measure of quality may not fully capture the complexity of the solution space.

In the following case we show a further example of multiple solution space, with different features. we consider a ring of cliques (RC) network with additional bridge nodes. This configuration introduces a new form of symmetry, where the network

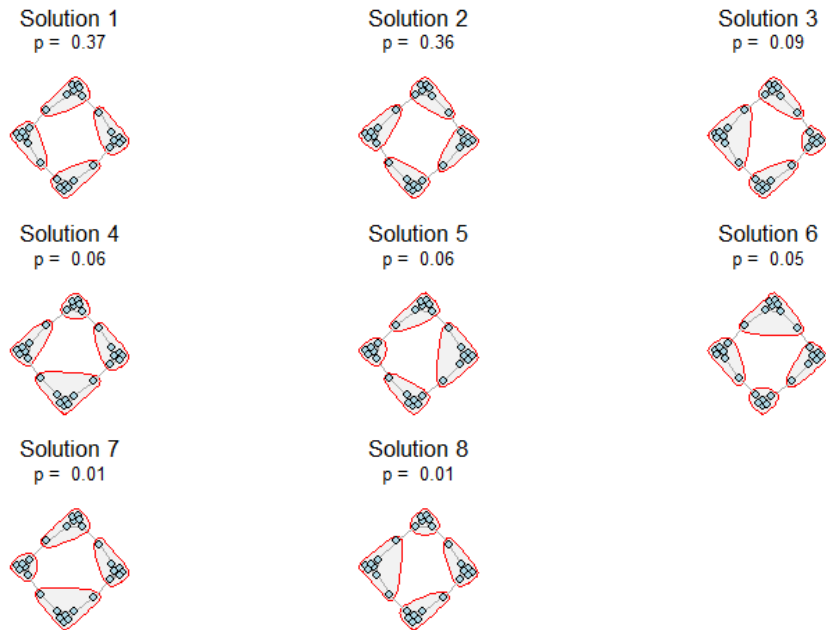


Figure 4.10: Solutions present in \mathbb{S}

can be partitioned in two equivalent ways: clockwise or counterclockwise. The solution space, in this case, consists of eight distinct solutions. Among these, two solutions dominate with equal probabilities of $p \approx 0.36$, while the remaining six solutions have much lower probabilities, with $p < 0.1$. Figure 4.10 illustrates this distribution.

his structure is well captured by the solution space. The two prominent solutions reflect the natural symmetry of the network, corresponding to the clockwise and counterclockwise partitioning of the ring. The minor solutions represent variations that are less aligned with the network's core structure, thus appearing far less frequently.

In this scenario, if the analytical framework requires a single solution, several approaches could be taken. One option is to apply a consensus method that incorporates all eight solutions, balancing the contributions of both the dominant and minor partitions. Alternatively, a more focused consensus approach could be adopted by restricting the analysis to the two most frequent solutions (Solutions 1 and 2), ensuring that the symmetry of the network is respected while simplifying the solution space. Figure 4.10 shows how the solution space evolves as trials progress, with the two dominant solutions quickly emerging as stable and the minor solutions remaining infrequent. Stability is typically reached at around $t=120$, with parameters $t_{\max}=200$ and $\tau=0.95$.

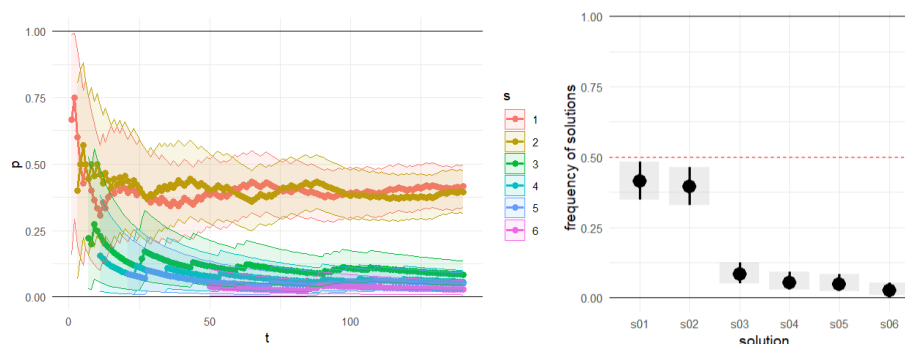


Figure 4.11: Evolution of the knowledge on solution space. Parameters: $t_{max} = 200$, $\tau = 0.95$. Result: \mathbb{S} is stable at $t = 130$ trials. There multiple solutions, divided in two groups.

This example underscores the flexibility of the solution space framework in capturing the true structure of the network. By considering multiple partitions, particularly in cases of symmetry, the analysis can better represent the underlying dynamics of the network. When the solution space contains symmetrically equivalent solutions, as in this case, consensus approaches can help in selecting or combining solutions, ensuring that important structural properties are preserved without over-simplifying the analysis.

Sparse

In the "sparse" solution space, we analyze a ring of cliques (RC) network with a central node and additional bridge nodes, which results in a highly complex structure with numerous potential partitions. Despite the network's underlying symmetry, no dominant solution emerges in this case. Instead, the solution space is characterized by a high number of distinct solutions, each occurring with very low probability. Specifically, the probability π for each solution is close to zero, and the number of solutions is nearly equal to the number of trials.

For this experiment, we limited the number of trials to $t=50$ for the sake of clarity in visualizing the figures, but even at $t=500$, the stability criterion is not met. Figure 4.12 shows the distribution of solutions, where all are equally unlikely and no single solution stands out.

The evolution of the solution space is depicted in Figure 4.12. Unlike the previous cases, the solution space does not stabilize, even with an increased number of trials. As more trials are conducted, new solutions continue to emerge, reflecting the complexity of the network's structure. This lack of stability is a defining feature of the sparse solution space, where the algorithm struggles to converge on a few stable solutions due to the inherent variability and multitude of possible partitions.

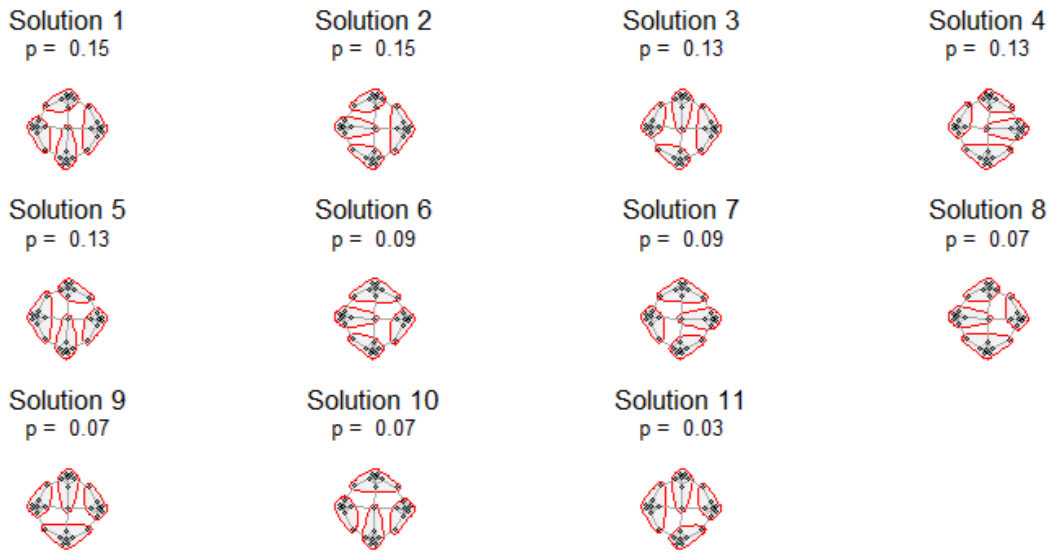


Figure 4.12: Solutions present in \mathbb{S}

Achieving a single, reliable solution in this scenario can be challenging due to the high variability in the solution space. One potential approach is to apply a consensus method across all solutions, combining the information from multiple partitions to create a more robust final result. Alternatively, simplifying the network by pruning unnecessary elements, such as peripheral nodes or bridges, could help reduce the number of possible partitions. After pruning, the network could be reanalyzed, potentially yielding a clearer and more stable solution space.

This example demonstrates that, in cases of high complexity where the solution space is sparse and no clear solution emerges, additional steps such as consensus building or network pruning are necessary to arrive at a useful interpretation of the network's structure. The Bayesian framework helps identify when the solution space is unlikely to stabilize, guiding further efforts to refine the analysis.

Empty

In the "empty" solution space, we analyze a random graph with no inherent community structure. When applied to such a graph, community detection algorithms still return partitions, but the partitions are entirely inconsistent across trials, with each partition being different from the others. What distinguishes the "empty" case from the "sparse" case is the degree of similarity between the solutions.

In the sparse case, even though many solutions are found, there is still some underlying community structure, no matter how fuzzy, leading to similarities between the partitions. This can be quantified using the Normalized Mutual Information

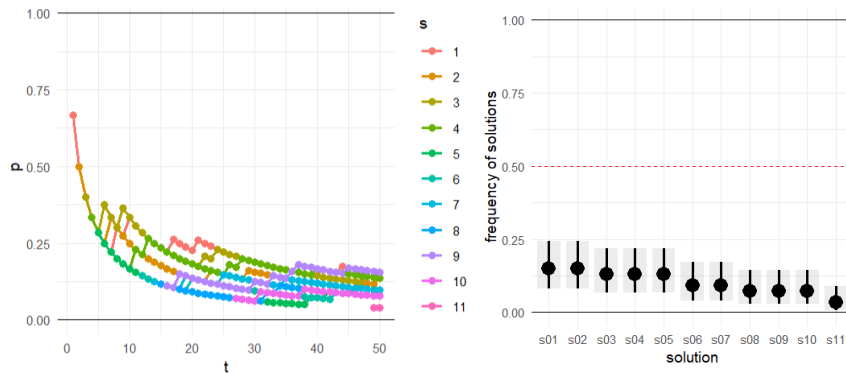


Figure 4.13: Evolution of the knowledge on solution space. Parameters: $\tau = 0.95$ and $t_{max} = 50$. The value has been chosen to improve clarity of the figure. The solution space is not stable and new solutions appear also at $t = 1000$.

(NMI) between pairs of solutions, which measures how similar the partitions are. In the sparse case, NMI scores between pairs of solutions typically remain above 0.5, indicating that the solutions share some common structure.

By contrast, in the empty case, the solutions generated by the algorithm are completely dissimilar, with NMI scores falling below 0.5. This lack of similarity indicates that the partitions are random and carry no meaningful information about the structure of the graph. Figure 4.14 shows this stark contrast between the sparse and empty cases, with similarity coefficients clearly separating the two types of solution spaces.

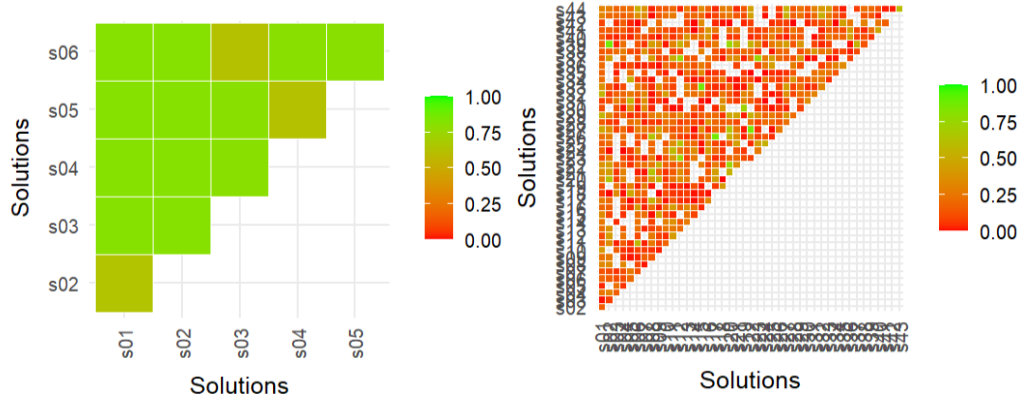


Figure 4.14: Similarity between solutions \mathbb{S} for the case of sparse (left) and empty (right).

Another useful metric for identifying an empty solution space is the mixing parameter μ , which measures the ratio of inter-community connections to intra-community connections as described in chapter 2. In this "empty" scenario, there is no structure in the graph, and thus, no valid partitions can be identified. As a

result, the solution space \mathbb{S} is effectively empty, meaning that none of the returned partitions can be considered valid or meaningful representations of the network. This is in contrast to the sparse case, where despite the complexity and variability, there are still some valid partitions emerging from the analysis. More examples of solution space are illustrated in the case studies in chapters 6 and 7.

4.2 Consensus Community Detection

The exploration of the solution space is essential for determining whether a unique solution exists. In cases where multiple solutions are present, it becomes necessary to identify one that is both valid and stable. An example will clarify the importance of this process. Consider the goal of understanding how knowledge flows to enable and foster innovation, as discussed in Chapter 1. Collaborations are modelled using networks, where communities, or cohesive groups, play a key role in analysis and interpretation. However, discovering communities is not the primary objective; instead, it is a means to better understand and interpret the network's community structure. Interpreting the results in a reliable and reproducible way requires that $\mathcal{A}(G, \rho) \rightarrow P$ that meets the following conditions:

- \mathcal{P} should be composed of $1 < k < n_v$ communities
- each $C \in \mathcal{P}$ should be valid according to the four criteria explained in section 3.2;
- \mathcal{P} should be insensitive to the specific formulation of the network within the given programming environment.
- P should not vary upon repeated executions of \mathcal{A} ;

The first two criteria can be addressed through a quality check on \mathcal{P} and C , flagging inconsistent results and excluding them from further analysis. Ordering bias and variability are mitigated by exploring the solution space, leaving only the issue of multiple valid solutions to be addressed.

One approach is to identify as "optimal partition" $\mathcal{P}_x \in \mathbb{S}$ that maximizes an objective function such as modularity. This approach naturally fits with modularity-based algorithm such as LV or LD, but is less consistent with algorithms as IM, WT and LP. This approach yields a single solution, but not a stable one, as it may be surpassed by subsequent iterations of the procedure. Moreover, in the case of degeneracy (section 2.4) there may be a large number of solutions with only minimal differences in modularity. Relying solely on modularity to choose one of these solutions can result in the loss of valuable information. Furthermore, there exists no clear correlation between modularity optima and any features relevant

for interpretation, such as the number of communities, as illustrated in Figure 3.1 (bottom row).

Consensus offers a more robust option to enhance stability. For example, as discussed in [50], distinct partitions obtained by repeated execution of \mathcal{A} are used to build a co-occurrence matrix, D , in which each entry d_{ij} signifies the proportion of partitions in which vertices i and j are clustered together. D is then interpreted as an adjacency matrix for a new network, representing the community structure. In the new network, edges below a chosen threshold p are pruned, and the process is repeated recursively until D is a block-diagonal matrix, where each block is interpreted as a community. The process is effective but has the disadvantage of requiring multiple iterations. Moreover, pruning can generate disconnected nodes (vertices that have all edges below the threshold p), hence a threshold $p = 0.6$ is recommended. To maintain network connectivity, disconnected nodes are aggregated into the neighbouring community with the highest weight. The algorithm's ability to identify communities of varying scales and its capacity to properly identify outliers is limited by this assumption, as discussed in section 3.3.

Another approach can be found in [38] and [39], which propose the Ensemble Louvain algorithm to find stable communities. As with the previous method, a co-occurrence matrix D is calculated, and communities are identified by pruning with a threshold $p = 0.9$. Selecting such a high threshold value returns more stable results without necessitating recursive iterations but has the drawback of overlooking outliers, i.e. all the nodes that fall above the threshold. Depending on the network topology and the objective of the analysis, outliers may be a negligible minority.

Other consensus approaches have been presented, to address specific issues. For example, [11] is focused on incomplete networks, and leverages a link-prediction strategy to infer missing intra-community edges and casting results with a consensus approach. Ensemble methods involve combining the outcomes of multiple community detection algorithms. One notable example is the ensemble method introduced by [15] that aims to identify overlapping and fuzzy communities.

The novel Consensus Community Detection (CCD) procedure introduced in this thesis can be applied to any community detection algorithm, to produce a stable representation of communities, improving the reliability and interpretability of results. Specifically, CCD addresses the four challenges outlined in section 4, dealing with the validity of detected communities, reducing variability across different algorithm runs, quantifying the residual variability, dealing with outliers, and mitigating the input-ordering bias.

While the variability of clustering results is widely explored in data science, its direct application to network community detection has been less emphasised in the

literature. Notably, the specific considerations regarding the handling of outliers and input-ordering bias in the context of community detection have been largely overlooked.

CCD is based on the assumption that **uncertainty is an inherent characteristic of community detection**, hence it should be carefully assessed and incorporated into the results.

For this reason, the results of a CCD algorithm produce an extended representation of a community as a vector of triplets node, label and uncertainty coefficient:

$$\tilde{\mathbf{C}} = \begin{pmatrix} (v_1, l_1, \gamma_1) \\ (v_2, l_2, \gamma_2) \\ \vdots \\ (v_n, l_n, \gamma_n) \end{pmatrix}$$

where l is the community label assigned to node v , and $\gamma \in [0, 1]$ is a coefficient that represents the uncertainty associated with the assignment of c_i .

$\gamma = 0$ indicates that the corresponding node is always co-occurring in the same community of at least one other vertex of the community. Higher values of γ indicate that the vertex was associated with different communities at each trial of the community detection.

The CCD approach builds on previous work but differs in three major aspects: addresses input-ordering bias, and introduces the novel *uncertainty coefficient* γ serving as a concise representation of residual variability at the node level, which can be subsequently leveraged for in-depth network analysis.

CCD provides a comprehensive framework to augment the efficacy of existing community detection algorithms, hence it maintains compatibility with legacy methods, enabling straightforward comparisons with prior analyses and established literature.

Given the issues discussed in this chapter and the exploration of the solution space in Algorithm 1, a general workflow for identifying a single, stable community structure is proposed. This workflow is built on the premise that variability in community detection outcomes is inherent and must be managed.

The algorithm accepts as input a graph G , a community detection algorithm \mathcal{A} , a maximum number of iterations t_{\max} , and a similarity threshold τ . The output consists of a final, stable partition \mathbf{P} , representing the detected community structure, and a vector of uncertainty coefficients which quantifies the uncertainty associated with each node.

The first step in the workflow involves exploring the solution space \mathbb{S} by executing the community detection algorithm \mathcal{A} multiple times, as outlined in Algorithm 1. Following this, a quality check is performed on \mathbb{S} : partitions are tested against the

validity criteria described in Section 3.2, and their pairwise similarity is measured. This ensures that the subsequent analysis focuses only on meaningful and well-formed partitions.

The final step processes the solution space \mathbb{S} according to the taxonomy defined in Section 4.1.b, which classifies \mathbb{S} into four cases:

- **Empty:** If no valid partitions are found, the algorithm returns NA, indicating no valid result is available.
- **Single:** If \mathbb{S} contains only one valid partition, that partition is returned as the final output.
- **Dominant:** If one partition clearly dominates the others, this dominant partition P_{dominant} is selected as the final community structure.
- **Multiple or Sparse:** If no dominant solution exists, the algorithm applies a consensus procedure to generate a consensus partition $P_{\text{consensus}}$, which represents the most stable community structure derived from the available solutions.

This workflow offers flexibility by adapting to various solution space scenarios, ensuring robustness in identifying community structures even when the solution space is complex or ambiguous. The workflow could be further refined by introducing a specific case for "sparse" solutions, such as performing a manual review of the partition similarity or intervening on the network structure—e.g., by pruning edges to reduce network density and reapplying the detection algorithm to seek a clearer solution.

The pseudocode for the complete complete community detection procedure is represented in Algorithm 4.2.

The previous algorithm calls for a function $CCD(\mathbb{S}, p)$, which refers to the method used to identify stable community structures in the case of multiple or sparse solution spaces .

The algorithm takes as input a solution space object \mathbb{S} , along with two thresholds: p , which determines whether two nodes should be considered part of the same community, and n_{min} , which sets the minimum size for grouping outliers as described in 3.3.

The first step of the CCD algorithm is removing from \mathbb{S} that marked as invalid. This is optional, but recommended as it ensures that only valid solutions are included, thereby improving the reliability of the consensus results.

Next, the algorithm computes a co-occurrence matrix D that tracks how frequently pairs of nodes appear in the same community across different solutions. Each solution is weighted according to its frequency in the solution space, and the

Algorithm 2: Community Detection

Input: Graph G , algorithm \mathcal{A} , t_{max} , τ , p **Output:** A single, stable partition $\tilde{\mathbf{P}}$ **Explore solution space:** generate \mathbb{S} as per algorithm 1**Quality check on \mathbb{S} :** identify valid partitions and similarity**Proceed according to solution space taxonomy:** **switch** *Proceed according to the type of \mathbb{S} do* **case "Empty" do**

| Output NA

end **case "Single" do** | Output the single solution P_{single} **end** **case "Dominant" do** | Output the dominant solution $P_{dominant}$ **end** **case "Multiple" or "Sparse" do** | Output $\tilde{\mathbf{P}} \leftarrow \text{CCD}(\mathbb{S}, p)$ **end****end**

matrix is updated by adding the appropriate weights whenever two nodes co-occur in the same community.

The third step involves identifying consensus communities. The matrix D is processed in sequence, starting from the first row. Nodes having a co-occurrence score greater than the threshold p are considered part of the same community. Typical values of the threshold are $p=0.6$ to incorporate outliers and obtain large communities, or $p = 0.9$ to highlight outliers. An uncertainty coefficient γ is calculated for each node, as the mean value of coefficients $D[i, j]$ within the group. If no other nodes meet the threshold, the node is marked as a singleton, representing an isolated node with no strong community ties.

Finally, the algorithm handles outliers. If the algorithm detects communities with fewer than n_{\min} nodes, those nodes are re-assigned to a default label and grouped together as outliers. This ensures that small, isolated communities do not skew the results or are incorrectly interpreted as significant community structures. To avoid grouping, n_{\min} should be set to 0.

The output of the algorithm is a data frame that associates the node identifiers, their assigned community labels, and uncertainty coefficients. This provides a comprehensive view of both the community structure and the confidence in each node's assignment, offering a robust and interpretable analysis of the network.

The pseudocode for the CCD algorithm is represented in Algorithm 4.2.

Algorithm 3: Consensus Community Detection

Input: Solution space object SSP, threshold for community detection p ,
threshold for grouping outliers, n_{min}

Output: Partition $\tilde{\mathcal{P}}$

1) Filter valid trials

SSP \leftarrow SSP such that SSP.valid == TRUE;

2) Calculate co-occurrence matrix

$\alpha \leftarrow$ weight of each solution $P \in S$;

$D \leftarrow$ 0 matrix of size $n_nodes \times n_nodes$;

foreach solution $P \in S$ **do**

foreach community C in trial P **do**

foreach node $n_i \in C$ **do**

foreach node $n_j \in C$ **do**

 Increment $D[i, j]$ by $\alpha[t]$;

end

end

end

end

3) Identify consensus communities

Initialize all nodes as not processed Initialize community label $i \leftarrow 0$

while there are unprocessed nodes **do**

$i \leftarrow i + 1$;

$\bar{D} \leftarrow$ subset of D representing all unprocessed nodes;

$C \leftarrow \{\text{nodes } u \mid \bar{D}[1, j] > p\}$

if $|C| > 1$ **then**

 Calculate uncertainty γ for each node in C

end

else

 Mark the node in C as a singleton Assign $\gamma \leftarrow 0.0$

end

 Assign community label i to all node(s) in C Assign community size n to
all node(s) in C Mark all nodes in C as processed

end

4) Handle outliers:

if $n_{C_{min}} > 0$ **then**

 Identify all nodes within communities C_i such that $n \leq n_{min}$ Re-assign
the community label $c_i \leftarrow 0$;

end

return Partition $\tilde{\mathcal{P}}$

4.3 Performance of CCD

In this section, we show the results of tests on CCD to address all the issues shown in 4, namely its ability to reduce the variability of results, assess uncertainty, identify outliers, and reduce the input-ordering bias. Finally, we evaluate the performance of CCD in identifying a known community structure in three cases: (1) Karate network; (2) a family of RC networks with a fixed μ , but varying numbers of communities, and (3) a family of LFR networks with varying value of the mixing parameter μ .

4.3.a Reduction of variability - parameter t

CCD operates through a repetitive process executed for a designated number of iterations, denoted as t . Our first test concentrates on evaluating residual variability in relation to t , a critical decision involving a trade-off between cost (increasing linearly with t) and performance. We utilize two metrics: the count of identified communities (k) and the similarity between all pairs of partitions, assessed with NMI.

The test is conducted on LFR benchmark networks with parameters as outlined in section 4 and a nominal mixing parameter of $\mu = 30$. CCD was implemented with $p = 0.6$, $q = 0.5$, and t values ranging from 5 to 500. Stability, measured as the similarity between pairs of partitions, ideally yields $S = 1.0$. Results, depicted in Figure 4.15, reveal that CCD significantly enhances stability compared to single trials, with stability increasing as t increases, gradually reducing dispersion and approaching the optimal value. Notably, each algorithm reaches a plateau at a distinct value of t . In practical applications, the choice of an optimal t involves a trade-off between result stability and computational resources, where the right balance depends on the interplay between the network characteristics, the chosen algorithm and the analysis objectives.

4.3.b Assessment of residual variability

To illustrate how CCD assesses uncertainty associated with nodes, we apply it to the Karate network mentioned in section 2. We use the LV algorithm with $t = 100$, and different values of the resolution parameter r to control the granularity of community structure.

Results are shown in Figure 4.16, where some nodes are labelled: H and A (leaders of the two main communities) and node 10 (that may belong to either community, depending on the chosen value of r and the random variation that characterizes each trial). The first three panels display the results of single trials, demonstrating how

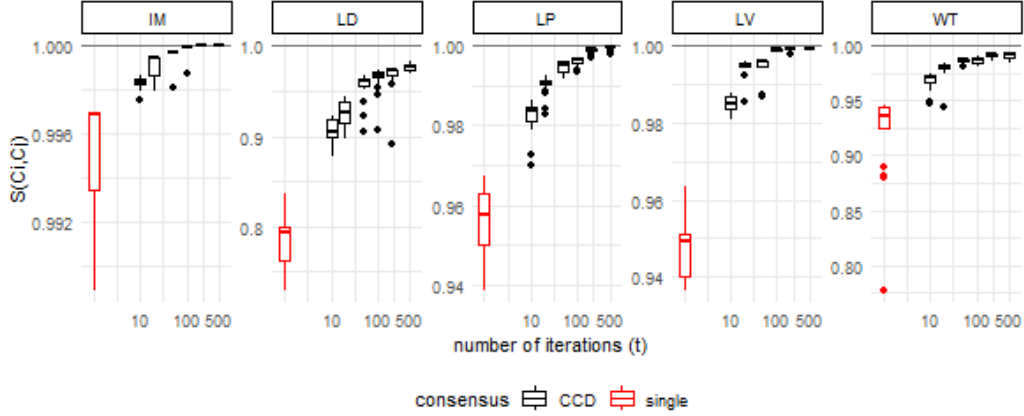


Figure 4.15: Stability of CCD results as a function of the number of iterations $t = (10, 20, 50, 100, 200, 500)$. Results of single trials $t = 1$ are highlighted in red. Test on a LFR network with $\mu = 0.3$, CCD parameters $p = 0.8$ and $q = 0.5$. Stability is measured by the similarity between pairs of solutions $S(C_i, C_j) = \text{mean}(NMI(C_i, C_j))$.

the number of communities k depends on the resolution parameter r . For example, in panel a) with $r = 0.5$ the result is $k = 2$; in panel b) with $r = 0.8$ there are two distinct results: $k = 3$ (in 61% of trials), and $k = 2$ (39% of trials). As per panel c), setting resolution $r = 1.0$ leads to $k = 4$. In the context of unsupervised machine learning, all the above results are equally valid. However, even when the value of the r is fixed, there is still significant variability that hinders interpretation. Panel d) shows how CCD can improve the interpretability: selecting $r = 0.5$, $p = 0.9$ and $q = 0.5$, produces a simple community structure with $k = 2$ and highlights node 10 as an outlier, with an uncertainty $\gamma = 0.75$, expressed by the color scale. Panel d) showcases a more nuanced application of CCD, where the resolution parameter assumes a different value at each trial, randomly selected in the range $[0.5, 1.0]$ which allows identification of $k = 3$ communities at different scales, and associates different levels of uncertainty to each.

Uncertainty is assigned at each node, but it can be summarised at the network level by the number of nodes with some degree of uncertainty or by the mean value of the uncertainty coefficient. Figure 4.17) shows both measures for a set of LFR with the characteristics presented in section 4; specifically the top row shows the fraction of nodes with $\gamma > 0$ and bottom row the median value of γ ; the shaded area is delimited by 10^{th} and 90^{th} percentile. In both cases uncertainty increases non-linearly with the mixing parameter. However the behavior is different according to the algorithm: in this example, IM identifies the communities with almost no variations for $\mu < 0.4$, then increases sharply. Other algorithms show a growing number of uncertain assignments at low values of μ , and plateau for $\mu > 0.3$.

Figure 4.18 shows a possible use of γ in unraveling network structure, in conjunc-

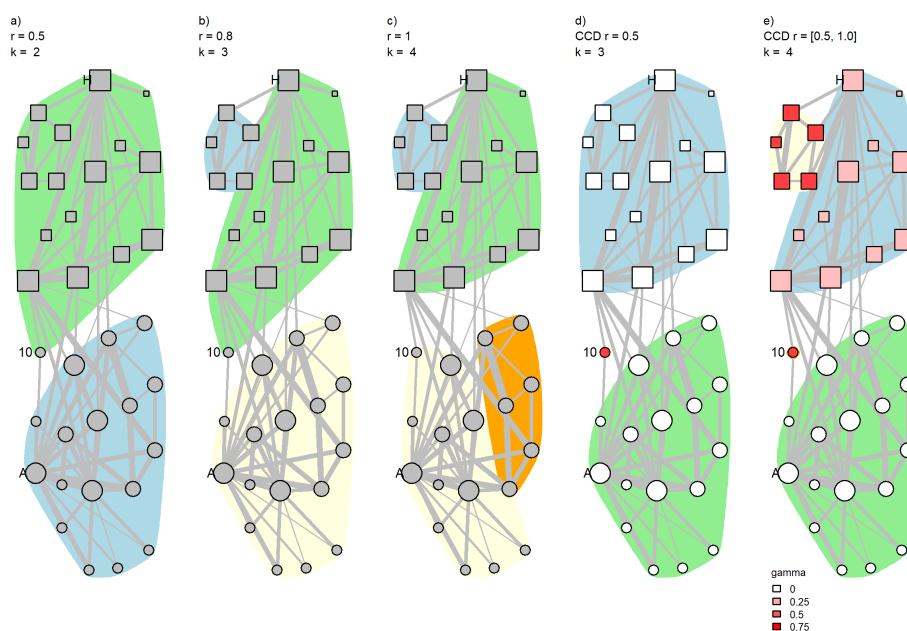


Figure 4.16: Example of CCD Zachary's Karate network (weighted). a) single trial of Louvain with resolution $r = 0.5$. b) single trial of LV, $r = 0.8$. c) single trial of LV, $r = 1.0$. d) CCD with $t = 100$ and $r = 0.5$ e) CCD with $t = 100$ and $r \in [0.5, 1.0]$. Uncertainty coefficient γ is available only for CCD.

tion with a centrality centrality measures - in this example k -coreness, a centrality measure introduced by [49]. Specifically, a k -core is a subgraph where all vertices are connected to at least k other vertices within that subgraph; the k -coreness of a node indicates the highest k -core that the node belongs to. The example is calculated on a LFR benchmark network with $\mu = 0.4$, and communities are detected CCD with parameters $t = 1000$, $p = 0.6$, and $q = 0.5$. The scatterplot depicts k -coreness against γ ; two examples of nodes with high uncertainty are highlighted by the arrows, and their respective neighborhood (at geodesic distance equal to 2) is depicted as subgraph. A single-node component is represented in the top left corner of the scatterplot (k -coreness = 0 and $\gamma = 1$).

Finally, a specific test is carried out to assess the ability of different algorithms to assign an appropriate value of γ . To ensure a reproducible example with a known expected value, the test is conducted on a family of RCs with clique sizes $s = 6$ and a number of cliques in the range $k_0 \in \{5, \dots, 100\}$. CCD was applied using $p = 0.8$, $q = 0.5$ and $t = 200$. The test is focused on the bridge nodes, i.e. nodes that connect two successive cliques within the ring, and can be expected to have $\gamma = 0.5$. Fig 4.19 shows that most algorithms behave well within a limited range of k_0 , and that for larger rings there are remarkable variations depending on the algorithm. IM and

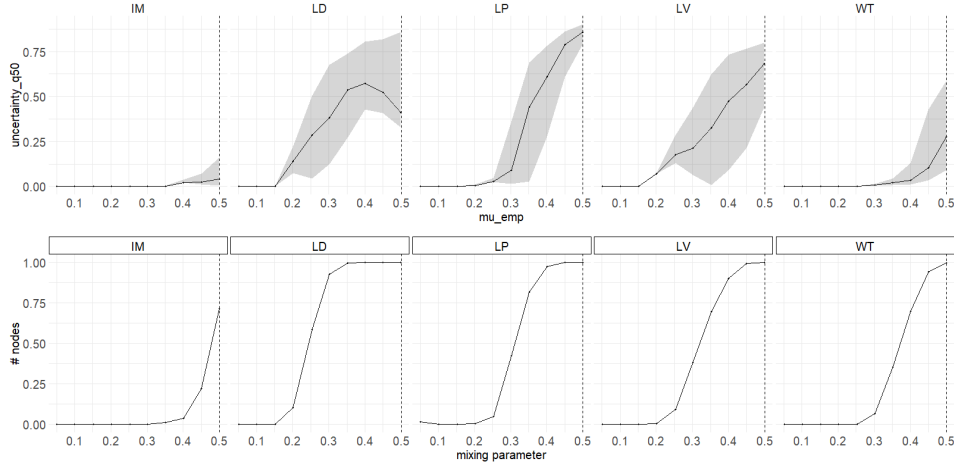


Figure 4.17: Assessment of uncertainty with CCD on a family of LFR benchmark networks. Top: median value of γ ; the shaded area is delimited by 10th and 90th percentile. Bottom: fraction of nodes with $\gamma > 0$.

LP produce very stable results even for $k_0 = 100$.

4.3.c Assessment of performance

In this section we evaluate the performance of CCD in identifying a known community structure, focusing on the ability to determine the number of communities and measuring the similarity between the inherent community structure and the outcomes of community detection.

The first test evaluates the ability of CCD to detect communities of varying sizes, on a family of LFR benchmark networks with parameters presented in section 4, Performance is assessed with two indicators: NMI (similarity between the identified communities and the built-in communities), and the normalized number of communities (k/k_0). CCD parameters are $t = 1000$, $q = 0.5$

Figure 4.20 compares the performance of the three different strategies to manage outliers discussed in section 3.3: for low μ , the curves overlap, indicating no significant deviations; however, as μ increases, differences emerge. Incorporating outliers ($p = 0.6$) leads to the best performance in terms of k/k_0 , but may hinder performance measured by NMI, especially with modularity-based methods LV and LD. On the other hand, highlighting ($p = 0.8$) generates several single-node communities, resulting in lower performance in terms of k/k_0 , but may offer an advantage in the interpretation of results. Finally, grouping ($p = 0.8$ and all outliers re-assigned to community 0) provides a trade-off between the previous options: it captures community structure (NMI comparable to previous case), still allows for

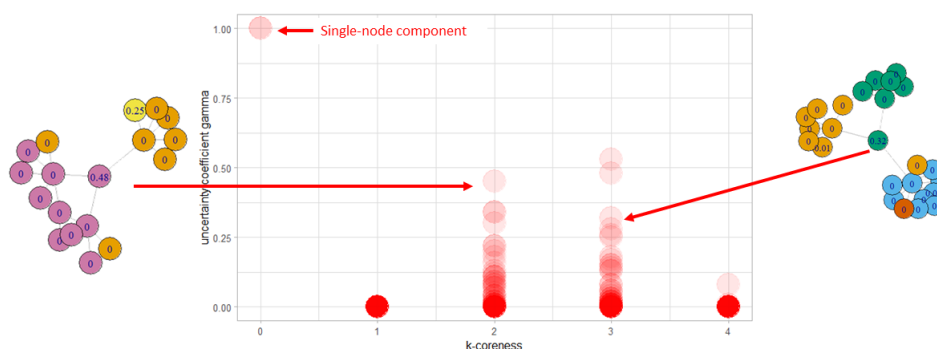


Figure 4.18: A coreness-uncertainty diagram on LFR network with $\mu = 0.40$. The subgraphs on the left and right of the diagram show the neighborhood of two nodes with high uncertainty.

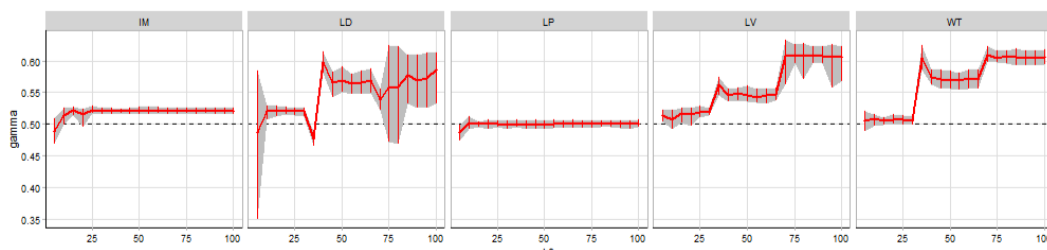


Figure 4.19: Uncertainty coefficient assigned by CCD to the bridge nodes of a family of RC benchmark networks. The expected value $\mu = 0.5$ is highlighted by the horizontal dotted line.

the identification of outliers and adds smaller errors to k/k_0 .

Figure 4.21 compares the performance of CCD (incorporating outliers, $p = 0.6$) with single trials and the recursive consensus community detection technique introduced by Lancichinetti et al. [50]. When measuring performance with NMI, consensus methods are outperforming single trials, especially as μ increases, although with different behavior depending on the algorithm. Performance measured with k/k_0 is comparable for WT and IM and diverges for the other methods as the fuzziness of the benchmark network approaches the limit value of $\mu = 0.5$.

The second test is focused on the effectiveness of identifying small, non-overlapping communities of the same size. The test is performed on a family of RC where k_0 varies between 5 and 100; CCD parameters are $p = 0.8$, $q = 0.5$, the network is shuffled and outliers are grouped according as discussed in 3.3.

Results are shown in Figure 4.22, representing the number of cliques k_0 versus the number of communities detected by CCD (red), recursive consensus (green), and single trials (blue). A dashed line shows the ideal result $k = k_0$. For low values of k_0 all methods perform well: the number of communities identified by the algorithm

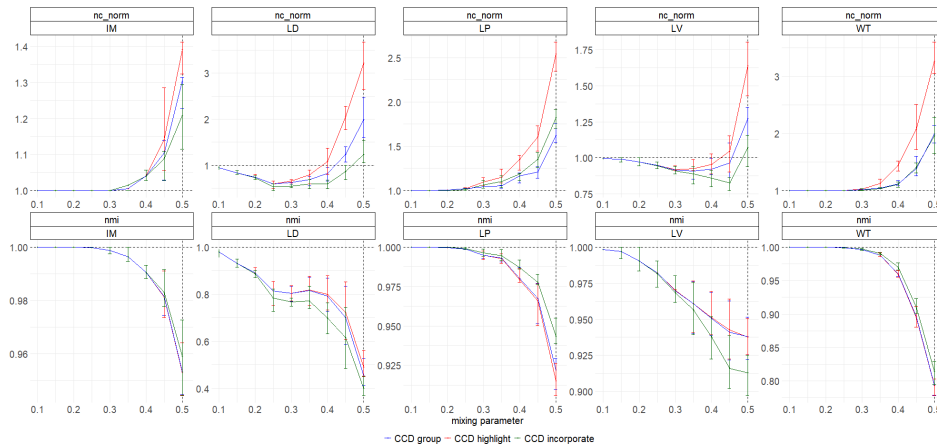


Figure 4.20: Performance of CCD on a family of LFR benchmark networks, using different strategies to manage outliers: group (blue), highlight (red) or incorporate (green).

k is equal (or very close) to the number of cliques in the network k_0 . However, as k_0 increases, most algorithms tend to agglomerate neighboring communities, resulting in $k < k_0$, with behavior depending on the algorithm. In all cases, CCD is more accurate than recursive consensus and single trials, even for small cliques $s = 3$ arranged in large rings up to $k_0 = 100$. In addition, CCD generally provides more stable results, as indicated by the vertical error bars in each plot, and allows the identification of outliers and quantitative assessment of uncertainty.

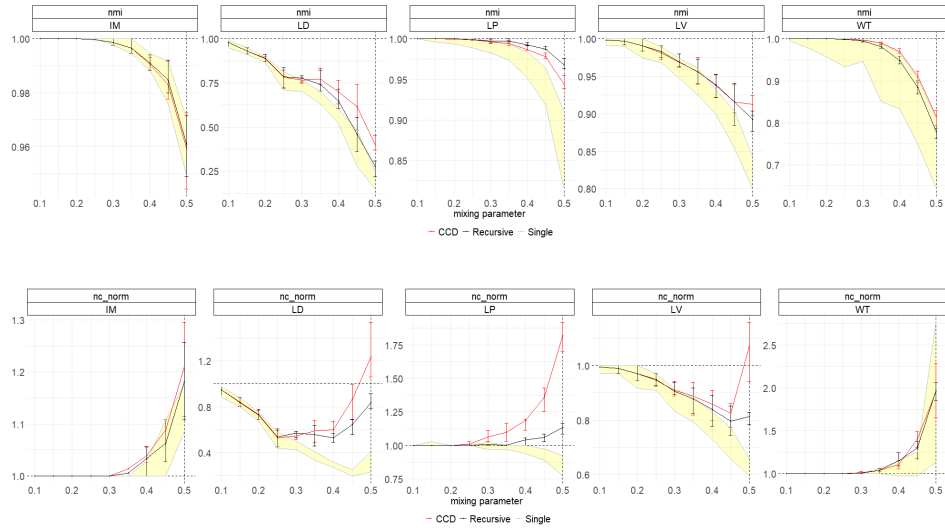


Figure 4.21: Performance of CCD on a family of LFR benchmark network. CCD (red), is compared to recursive consensus (black) and single trials (yellow).

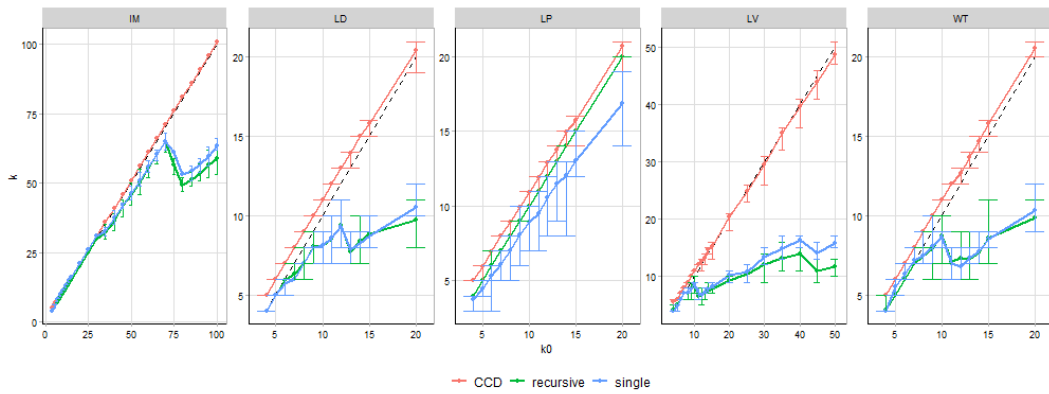


Figure 4.22: CCD results on a family of RC, compared to recursive consensus and single trials.

R-package 'communities'

This chapter introduces the R package "communities," which was developed as part of the research work. The package provides tools for the analysis and visualization of communities within networks, building on the functionality of the R-igraph library and incorporating the methods discussed in previous chapters. Its key features include the generation of benchmark networks, exploration of solution space, consensus community detection, analysis of community structure and enhanced visualization.

5.1 Functions to generate benchmark networks

5.1.a Function: `make_clique`

The `make_clique` function generates a fully connected subgraph (i.e., a clique) of a specified size and assigns a community label to all nodes in the clique. This function is used by `make_ring_of_cliques()`. The function constructs a clique using a given number of nodes and allows the assignment of a common community label to those nodes. The returned graph is undirected and can be combined with other cliques or networks. The function accepts the following arguments:

`clique_size` An integer representing the number of nodes in the clique. All nodes will be fully connected to each other.

`comm_label` A numeric or character value used to assign a community label to the nodes within the clique. The label will be stored in the

Return value: An undirected igraph object representing a clique, where all nodes are connected. Each node will have a `community` attribute assigned according to the `comm_label` parameter.

Usage:

```
# Create a clique of size 5 with community label 1
clique <- make_clique(clique_size = 5, comm_label = 1)
```

5.1.b Function: make_ring_of_cliques

The `make_ring_of_cliques` function generates a graph composed of multiple fully connected subgraphs (cliques) arranged in a ring. Optionally, "bridge nodes" can be added between cliques, and a central node can be added that connects to all cliques. Rings of cliques are useful for testing community detection algorithms. By varying the parameters, one can create toy examples of controlled complexity, ranging from trivially simple to particularly challenging. **Arguments:**

num_cliques An integer specifying the number of cliques to include in the ring.

clique_size An integer specifying the number of nodes in each clique.

add_center A logical value indicating whether to add a central node that connects to all cliques. Default is TRUE.

add_bridges A logical value indicating whether to add bridge nodes between adjacent cliques in the ring. Default is TRUE.

Value: the function returns an `igraph` object representing the ring of cliques. Each node is assigned a community label, with cliques labeled as C1, C2, ..., bridge nodes labeled as B1, B2, ..., and the central node labeled as A. The graph is undirected, and all edges have a weight of 1.

Usage:

```
# Create a ring of 4 cliques, each with 5 nodes,  
# with both bridges and a central node  
ring_of_cliques <- make_ring_of_cliques(  
  num_cliques = 4,  
  clique_size = 5,  
  add_center = TRUE,  
  add_bridges = TRUE)  
  
# Create a ring of 3 cliques, each with 4 nodes,  
# without bridges or a central node  
ring_of_cliques_no_center <- make_ring_of_cliques(  
  num_cliques = 3,  
  clique_size = 4,  
  add_center = FALSE,  
  add_bridges = FALSE)
```

5.2 Functions for solution space and quality check

5.2.a Function: `solutions_space`

The `solutions_space` function generates the solution space for a given community detection algorithm by running multiple trials. Each solution is assessed based on various performance metrics. The function accepts the following arguments:

g An `igraph` object representing the network.

algorithm A string specifying the community detection algorithm to be used.

trials The number of trials to generate different solutions. Default is 100.

Return value: A list containing the solution space, including the community membership matrix `ssp$M` and associated dataframe `ssp$data`.

Usage:

```
# Generate solution space using a specified algorithm  
g <- make_ring_of_cliques(3, 5)  
solution_space <- solutions_space(  
  g,  
  algorithm = "LV",  
  trials = 50)
```

5.2.b Function: `quality_check`

The `quality_check` function calculates several quality metrics to evaluate the robustness and integrity of the detected communities in the network. This function updates the results generated by the `solution_space` function. The function accepts the following arguments:

`solution_space` A vector of community labels assigned to each vertex in the graph.

`g` An `igraph` object representing the network to be analyzed.

Return value: A dataframe containing various quality metrics for the detected communities.

Usage:

```
# Perform a quality check on detected communities
g <- make_ring_of_cliques(3, 5)
ssp <- solutions_space(g, algorithm = "LV", trials = 50)
ssp <- quality_check(ssp)
```

5.2.c Function: `plot_sol_space`

The `plot_sol_space` function visualizes the solution space for a community detection algorithm. This plot helps users identify patterns, clusters, or optimal solutions based on various performance metrics. The function accepts the following arguments:

`solution_space` A matrix or dataframe representing the different solutions (rows) and their associated metrics (columns).

`title` A string specifying the title of the plot. Default is "Solution Space".

Return value: A plot visually representing the solution space, where each point corresponds to a solution with its performance metrics.

Usage:

```
# Example
g <- make_ring_of_cliques(3, 5)
ssp <- solutions_space(g, algorithm = "LV", trials = 50)
ssp <- quality_check(ssp)
plot_sol_space(ssp)
```

5.2.d Function: `empirical_mu`

The `empirical_mu` function calculates the mixing parameter "mu" for a given network, which represents the proportion of edges that exist between different communities. This parameter is often used in community detection algorithms to quantify how mixed or modular a network is. The function works by ensuring that all edges have weights (defaulting to 1.0 if no weights are provided). It calculates the number of edges that connect nodes from different communities and computes the proportion of such edges relative to the total number of edges in the graph. The function accepts the following arguments:

g An `igraph` object representing the network. The edges should ideally be weighted, but if not, all edge weights are set to 1.0 by default.

community_labels A vector of community labels, ordered according to the vertices in `g`. These labels define the community membership of each node in the graph.

Return value: A numeric value representing the empirical mixing parameter (`mu`), which is the ratio of inter-community edges to the total number of edges.

Usage:

```
# Create a simple graph and calculate the mixing parameter
g <- make_ring_of_cliques(3, 5)
community_labels <- V(g)$community
mu <- empirical_mu(g, community_labels)
print(mu)
```

5.2.e Function: `internally_connected`

The `internally_connected` function calculates how internally connected each community in the network is. It returns the number of connected components for each community, which helps evaluate the structural integrity of the communities. The function accepts the following arguments:

g An `igraph` object representing the network to be analyzed.

community_labels A vector of community labels, ordered according to the vertices in `g`. These labels define the community membership of each node.

Return value: A numeric vector where each entry corresponds to the number of connected components within a community.

Usage:

```
# Example of calculating internal connectivity
g <- make_ring_of_cliques(3, 5)
community_labels <- V(g)$community
internal_conn <- internally_connected(g, community_labels)
print(internal_conn)
```

5.3 Functions for consensus community detection

5.3.a Function: `co_occurrence`

The `co_occurrence` function computes a normalized co-occurrence matrix based on the solution space provided by the input object `ssp`. The output matrix `D` indicates the extent to which pairs of nodes appear together in the same community across different trials.

The function filters out invalid trials based on the `valid` column in the `ssp$data` dataframe. It then computes the co-occurrence matrix `D` by iterating through each trial and identifying pairs of nodes that belong to the same community. The contribution of each trial to the co-occurrence matrix is weighted by the median values in the `ssp$data` dataframe, normalized by the total sum of medians.

For each valid trial, the function identifies the nodes that belong to the same community and increments their co-occurrence count. The co-occurrence matrix `D` is symmetric, reflecting the fact that if one node co-occurs with another, the reverse is also true. The diagonal entries are set to 1 to indicate perfect self-co-occurrence for each node. The function accepts the following arguments:

ssp A list containing the solution space, including the community membership matrix `M` and associated dataframe, as generated by the `solution_space` function.

Return value: A symmetric matrix `C0` where each entry represents the weighted co-occurrence count of node pairs across all trials.

Usage:

```
g <- make_ring_of_cliques(3, 5, add_bridges = TRUE)
ssp <- solutions_space(g, algorithm = "LV", trials = 50)
ssp <- quality_check(ssp)
D <- co_occurrence(ssp)
```

5.3.b Function: `consensus_communities`

The `consensus_communities` function identifies consensus communities from a co-occurrence matrix, grouping nodes into communities based on a threshold `p` for pairwise co-occurrence. It also calculates an uncertainty coefficient (`gamma`) for each node, reflecting how confidently each node belongs to its community. The function accepts the following arguments:

D A symmetric co-occurrence matrix where each entry represents the pairwise co-occurrence of nodes across multiple trials.

p A numeric threshold for defining communities. Nodes are considered to be in the same community if their pairwise co-occurrence value is greater than `p`.

group_outliers A logical value indicating whether single-node communities (outliers) should be grouped together. Default is `FALSE`.

verbose A logical value indicating whether to print detailed progress information. Default is `FALSE`.

Return value: A dataframe containing the following columns:

- `name` The name of each node.
- `cons_comm_label` The consensus community label assigned to the node.
- `gamma` The uncertainty coefficient for each node, calculated as $1 - \text{mean}(d_i)$ over all nodes that co-occur at least once in the same community.
- `comm_size` The size of the community to which the node belongs.
- `single` A boolean indicating whether the node is part of a single-node community (outlier).

Usage:

```
g <- make_ring_of_cliques(3, 5, add_bridges = TRUE)
ssp <- solutions_space(g, algorithm = "LV", trials = 50)
ssp <- quality_check(ssp)
D <- co_occurrence(ssp)
consensus_partition <- consensus_communities(
  D,
  p = 0.5,
  group_outliers = FALSE)
```


5.4 Functions for analysing community structure

5.4.a Function: `make_community_network`

The `make_community_network` function constructs a new network where each node represents a community within the original network. Edges between nodes represent the aggregated connections between those communities.

The function accepts the following arguments:

- g** An `igraph` object representing the original network to be analyzed. The network must include the following attributes: `V(g)$community`, an integer vector representing the community assignment for each node, and `E(g)$w` A numeric vector representing edge weights. If the network is unweighted, set `E(g)$w` to 1.0 for all edges.

Return value: An `igraph` object representing the community network with the following attributes:

- `V(Gc)$membership` Community labels corresponding to the ones in the original network.
- `E(Gc)$w` Edge weights representing the sum of edge weights between communities in the original network.
- `V(Gc)$size` The number of nodes from the original network `G` that belong to each community in `Gc`.

Usage:

```
# Create a community network from the original iGraph object
community_network <- make_community_network(g)
```

5.4.b Function: `comm_label_as_strongest`

The `comm_label_as_strongest` function assigns community labels to nodes based on the strongest edge (i.e., the highest-weight edge) connecting each node. This approach helps assign nodes to communities based on their most significant interaction. The function accepts the following arguments:

- g** An `igraph` object representing the network.

Return value: A vector of community labels for each node in the graph.

Usage:

```
# Example of assigning community labels based on strongest edge
community_labels <- comm_label_as_strongest(g)
```

5.5 Functions for visualization

5.5.a Function: `plot_solutions`

The `plot_solutions` function generates plots of all the solutions in the solution space of a given network. Plots are arranged in a grid layout, allowing for the visualization and comparison of solutions in a single frame. Each solution is plotted using the Fruchterman-Reingold layout to ensure consistent node positioning across all plots. Communities in each solution are highlighted based on the membership matrix `sol_space$M`. If the `device` is set to "png", the function saves the plot as a .png file, otherwise, the plot is rendered on the screen.

Arguments:

g An `igraph` object representing the network graph to be plotted. Each node corresponds to an entity, and edges represent relationships between them.

sol_space A list containing the solution space with the following components:

data A matrix, where each row corresponds to a distinct solution (e.g., a community detection result).

M A matrix of node memberships for each solution. Each column corresponds to a different solution, and each entry specifies the community to which a node belongs in that solution.

device A string indicating where the plot should be rendered. Possible values are "screen" (default) to display the plot, or "png" to save the plot to a .png file.

filename A string specifying the filename for saving the plot if `device = "png"`. The default is NULL.

width An integer specifying the width of the .png file in pixels. The default value is 1600.

height An integer specifying the height of the .png file in pixels. The default value is 1600.

res An integer specifying the resolution of the .png file in DPI. The default value is 300.

Return value: NULL. The function does not return any values. It creates plots either on the screen or in a .png file, depending on the specified device parameter.

Usage:

```
# plot to screen:
plot_solutions(g, sol_space)

# save the plot as a PNG file:
plot_solutions(g,
               sol_space,
               device = "png",
               filename = "solutions.png")
```

5.5.b Function: `layout_distance_comm`

The `layout_distance_comm` function calculates a layout for graph visualization that emphasizes the distances between communities. This layout helps visually differentiate between communities based on their structure and connections. The function accepts the following arguments:

g An `igraph` object representing the network.

membership A vector of community memberships.

eps A small threshold to determine proximity between nodes.

Return value: A matrix of node positions for plotting the network.

Usage:

```
# Example layout based on distances between communities
distances <- layout_distance_comm(g, membership, eps = 0.15)
```

Innovation patterns within a regional economy

This chapter is largely taken from the paper "Innovation patterns within a regional economy through consensus community detection on labour market network" [59].

Universities and research centres play a crucial role in generating and disseminating knowledge through education, research, spin-offs, technology transfer and participation in open innovation processes, as discussed in chapter 1. While connections between companies have been widely analysed through the concept of "clusters" based on spatial proximity, industrial similarity, or competition [75], this chapter introduces another knowledge dissemination mechanism: worker mobility between employers.

The use of labour market data to study inter-links between companies is based on the observation that when employees change jobs, they move to another employer geographically close, requiring similar skills and offering better conditions [7]. The analysis can be global, such as [68], which uses labour market data from the social network LinkedIn, or regional, such as [54], which use data from Italy's regional labour market observatories.

In this thesis, anonymized data from the Friuli Venezia Giulia labour market offers the opportunity to study knowledge transfer within the region with a network-based analysis. Broadly, the workflow starts by constructing a network, examining its fundamental structure through components, centrality indicators, and communities. Given that multiple valid solutions may exist within the solution space, a consensus approach is employed, with uncertainty coefficients helping to identify key organizations.

The results highlight the prominent role of research institutions in Friuli Venezia Giulia's innovation landscape. These institutions lead extensive communities within the network and maintain strong connections with industrial organizations, under-

scoring their significance in regional innovation dynamics.

6.1 Innovation in Friuli Venezia Giulia region

The case study presented in this chapter is focused on Friuli Venezia Giulia because both Area Science Park and the University of Trieste are interested in understanding the region's innovation dynamics, and a relevant dataset on the regional labour market is available.

Friuli Venezia Giulia (FVG) is a region in northeastern Italy, bordered by Austria, Slovenia, and the Adriatic Sea. Despite its relatively small size, the region plays a significant role in Italy's economy and innovation landscape.

The regional economy is driven by a dynamic ecosystem of small and medium-sized enterprises (SMEs) known for their innovation and high-value-added activities. These SMEs form the backbone of the regional economic structure, complementing the presence of several large, globally recognized enterprises. Notable companies include Danieli, a leader in steelmaking technology, and Fincantieri, one of the world's largest shipbuilding groups. These firms not only generate substantial economic activity but also drive innovation through advanced research and development. Other prominent players such as Generali Italia, Electrolux, and Biofarma Group further bolster the region's reputation as a hub of industrial and technological excellence.

Despite its modest population and geographic size compared with other Italian regions, FVG exerts an influence in innovation. The region places a strong emphasis on research and development, particularly in advanced manufacturing, biotechnology, and information technology. This focus is supported by a network of research institutions and universities, fostering a fertile environment for scientific and technological advancement.

Its status as an autonomous region allows for tailored support of economic development and innovation initiatives. A prime example of this is the Scientific and Innovation System of Friuli Venezia Giulia (SiS FVG), a collaborative initiative designed to enhance the region's scientific and technological capacities. SiS FVG is supported by partnerships between the Friuli Venezia Giulia Autonomous Region, the Italian Ministry of Foreign Affairs, and the Italian Ministry of University and Research, aiming to promote socio-economic growth through innovation. Another example of the impact of regional autonomy and border position is the construction of relevant cross-border innovation initiatives, such as the NAHV project discussed in chapter 7.

SiS FVG includes a diverse array of institutions that collectively contribute to the region's robust scientific and innovation ecosystem. Among its members are

the University of Trieste, the University of Udine, the International School for Advanced Studies (SISSA), national research institutions such as Area Science Park, the National Institute of Oceanography and Experimental Geophysics (OGS), the regional sections of Italian National Research Council (CNR) National Institute for Astrophysics (INAF), and National Institute of Nuclear Physics (INFN). Additionally, international research institutes like Elettra - Sincrotrone Trieste S.C.p.A, and the International Centre for Genetic Engineering and Biotechnology (ICGEB), the Abdus Salam International Centre for Theoretical Physics (ICTP), TWAS - The Academy of Sciences for the Developing World are integral to the system.

According to the European Innovation Scoreboard (EIS) [33], Friuli Venezia Giulia ranks among the higher-performing regions in Europe, classified as a Strong Innovator. EIS is an annual report that offers a comparative assessment of the innovation performance of EU Member States, other European countries, and regional neighbors. Its objective is to assist countries in identifying areas for improvement and fostering innovation policies.

The region's smart specialization areas include agrifood and bioeconomy, metal mechanics and integrated housing systems, smart health, maritime and shipbuilding, as well as cultural and creative industries and tourism.

6.2 Data and methodology

The labour market refers to the supply and demand for labor, where employers seek to hire workers and individuals offer their skills and services in exchange for wages. It encompasses all economic activities related to employment, such as hiring, job transitions, and contract terminations. In this context, labor market data provide a comprehensive view of employment dynamics, capturing both the creation and cessation of jobs across various sectors.

Labour market data encodes the information as *events* that can be either the beginning of a new employment contract, or its termination. Each event is associated with a date, an employee, an employer, a professional profile and a location. The full dataset includes 1155342 events involving 74317 local units of companies of all sectors and sizes, as well as universities and research centres, that have either started or terminated an employment contract in the Friuli Venezia Giulia region between 2014 and 2021.

Before analysis, the raw data requires cleaning and completion (e.g., adding implicit contract terminations). It is also processed to standardize certain aspects, such as replacing the employer name with the actual workplace in cases involving employment agencies.

Using the entire network of all professional groups would offer limited insight

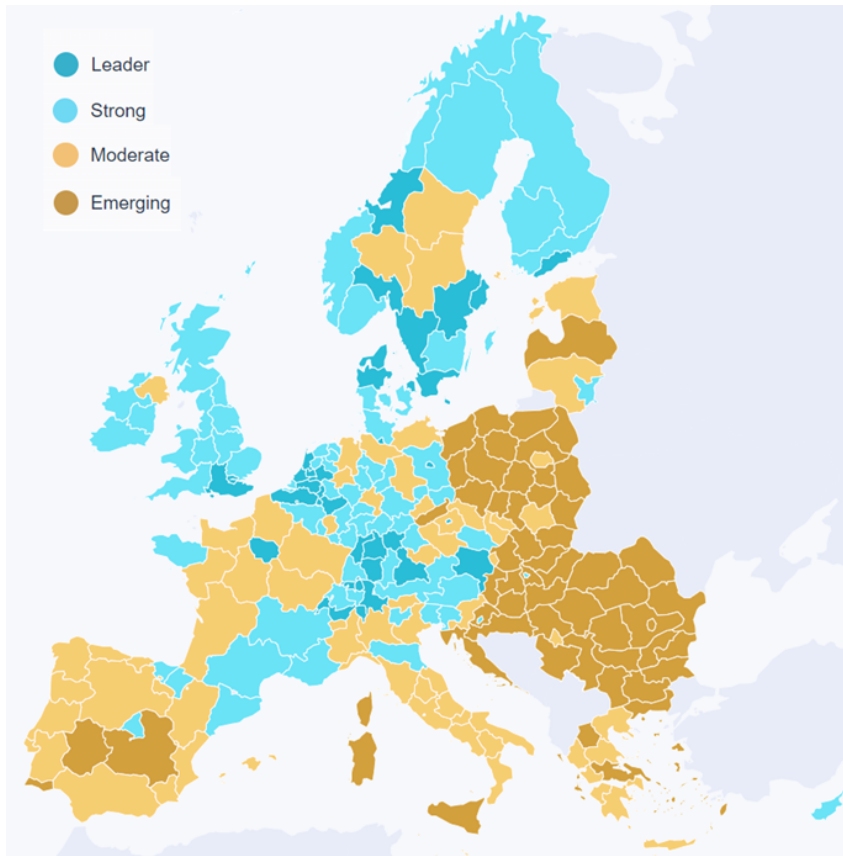


Figure 6.1: A map of European regions classified according to the European Innovation Scoreboard. The EIS categorizes countries and regions into four groups based on their innovation performance relative to the EU average. Innovation Leaders exhibit performance above 125% of the EU average. Strong Innovators score between 100% and 125%, Moderate Innovators range from 70% to 100%, and Emerging Innovators fall below 70%.

into innovation. The assumption is made that certain professional groups are more relevant to "knowledge transfer. To address this, a subset of professional groups is selected based on the International Standard Classification of Occupations [28], a system developed by the International Labour Organization (ILO) to classify and organize jobs within a standardized framework, to allow comparison of occupational statistics across countries and regions. Specifically, the subset is based on ISCO 2008 classification, and the selected professional groups are ISCO-21 (science and engineering occupations) and ISCO-25 (information and communication technology occupations). This results in a refined dataset containing approximately 60,164 events, involving 1890 employers and 16474 employees.

The network is built following the concepts discussed in Chapter 2. Vertices encode employers and edges encode the transition of an employee P from employer

A to employer B .

Transitions are assigned a weight which represents the relevance of the connection between A and B . The basic option is to assign a weight $W = 1.0$ to each transition; although this leads to valid results, we argue that it does not exploit the potential of the data. In this study, the weights are assigned under the assumption that the experience gained by P while working for A is transferred to B . Our data cannot capture the intrinsic economic value of each transfer, so we have chosen to approximate it with a non-linear parameter W . Let D_P^A be the duration of the contracts of P with A , D_P^B be the duration of the contracts of P with B (both expressed in years), and $maxW$ be a threshold that model the fact that experience gained in previous workplaces is no longer relevant. Our analysis assumes that $W = \min(D_P^A, D_P^B, maxW)$ where $maxW = 5.0$.

The resulting network, after simplification through the removal of loops, multiple edges, and the pruning of isolated vertices, consists of 1084 nodes (representing employers) and encodes 1641 transitions as edges. The next steps in the analysis involve analysis of **components**, **centrality** measures and **community** detection.

6.3 Results and discussion

6.3.a analysis of components

Nevertheless, a dominant component emerges, consisting of 734 nodes, which accounts for the majority of the network's actors. The following figure illustrates the distribution of these components. In the main component, all nodes are interconnected, with some significantly larger, reflecting their higher prominence within the network. The remaining components are notably smaller; remarkably, the majority of components (128) consist of only 2 nodes.

The division into components is crucial for understanding the distribution of knowledge flows for innovation across the network. Since information flows along the network's edges, it remains confined within the boundaries of each component. Therefore, organizations within larger components, particularly the main one, are better positioned to engage in knowledge exchange and collaboration, enhancing their potential for innovation. In contrast, organizations that belong to smaller, isolated components are effectively excluded from the main innovation flows.

6.3.b centrality measures in main component

Centrality measures are the second step for understanding the role and influence of organizations within the network. Two key centrality measures used in this context are coreness and strength, as defined in Chapter 2

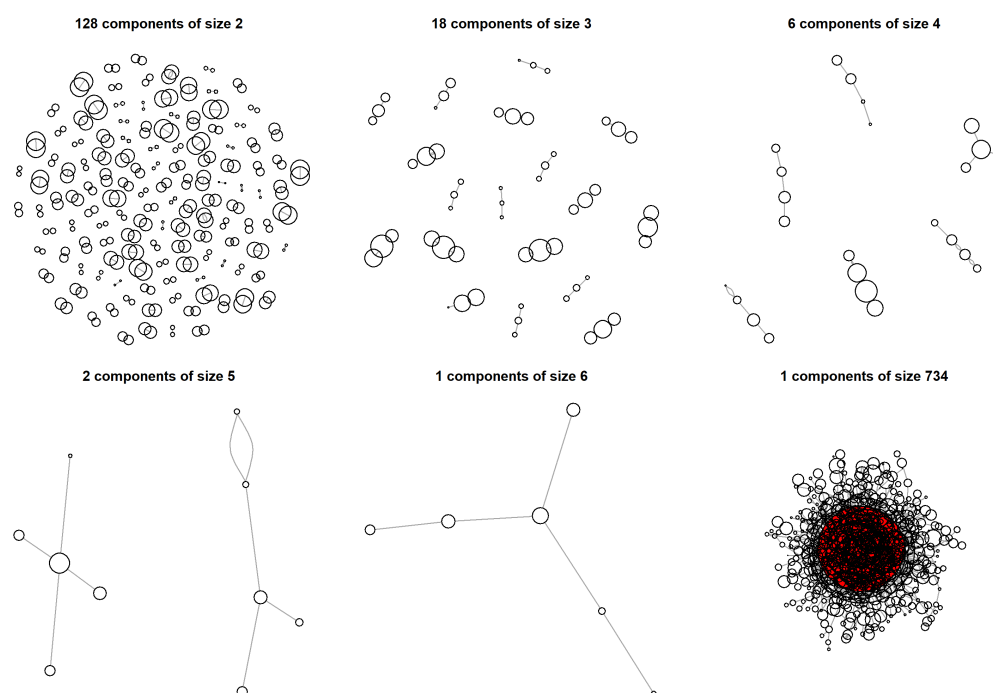


Figure 6.2: Components of the Labour Market Network. The network consists of 1084 nodes and 1641 edges, with the main component comprising 734 nodes. Node size is proportional to strength. Information and knowledge flow within individual components. Research organizations, highlighted in red, are all located within the main component.

The plot 6.3 visually represents each organization in the labor market network as a dot. The x-coordinate is the logarithm of the organization's strength, capturing the intensity its connections. The y-coordinate represents coreness, indicating an organization's position in the network's core-periphery structure. It is important to note that strength in this model is represented on a logarithmic scale: strength values below zero correspond to $10^0 = 1$, meaning these organizations have exchanged experience worth less than 1 year.

The plot is divided in two quadrants, on the right the main component, and on the left all other components. The analysis reveals that nodes belonging to minor components have both low coreness and low strength, indicating that these organizations are less influential and have little access to the main knowledge flows, at least from the perspective of this network model. This isolation likely limits their access to critical information, collaborations, and opportunities available to organizations in the main component.

All research organizations in the network show high coreness and strength, indicating their central position in the flow of knowledge. Their strong connections

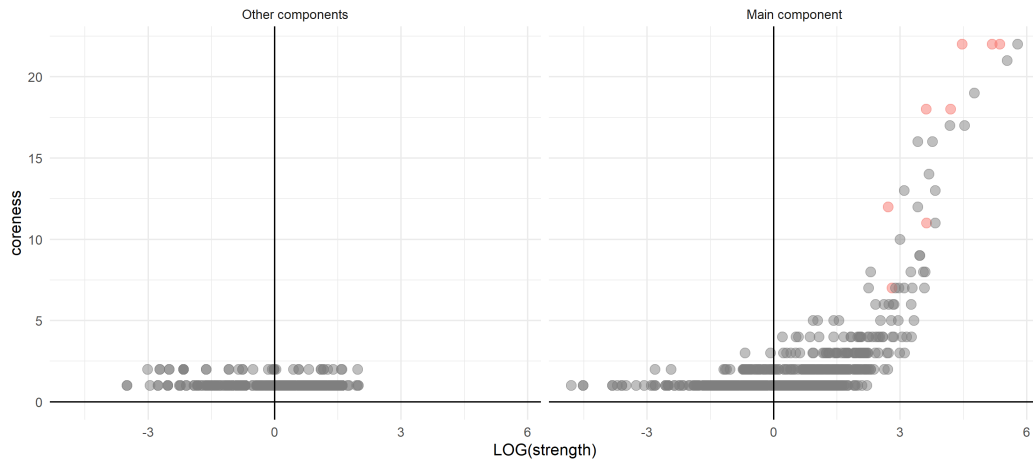


Figure 6.3: Strength vs. coreness for organizations in the labor market network. The x-axis shows the logarithm of strength, and the y-axis represents coreness. Organizations in the main component are on the right, while those in minor components are on the left. The red dots refer to universities and research centres.

likely enable them to play a key role in both disseminating and acquiring knowledge.

6.3.c community detection

The third step of the analysis involves community detection, which identifies cohesive groups within the network—specifically, groups of organizations that have participated in knowledge transfer through the movement of employees.

As discussed in Chapter 4, community detection algorithms should not be expected to produce a single definitive partition of the network but rather one of many possible partitions. Each partition can be viewed as a point within the broader solution space, representing a different possible configuration of communities within the main component. Hence a key step of the analysis is to explore the solution space, and identify a single solution through consensus. Outliers are present in this case, and two alternative strategies are showcased: *gorup* and *highlight*, as per Section 3.3.

The selected algorithm is LV (as per section 2). Since the solutions vary with each run, the solution space is explored over 100 iterations, yielding 99 independent solutions. To refine the analysis, pruning can be applied—for instance, removing weaker edges sharpens the identification of communities while preserving the most relevant ones. This process involves removing edges below a certain threshold,

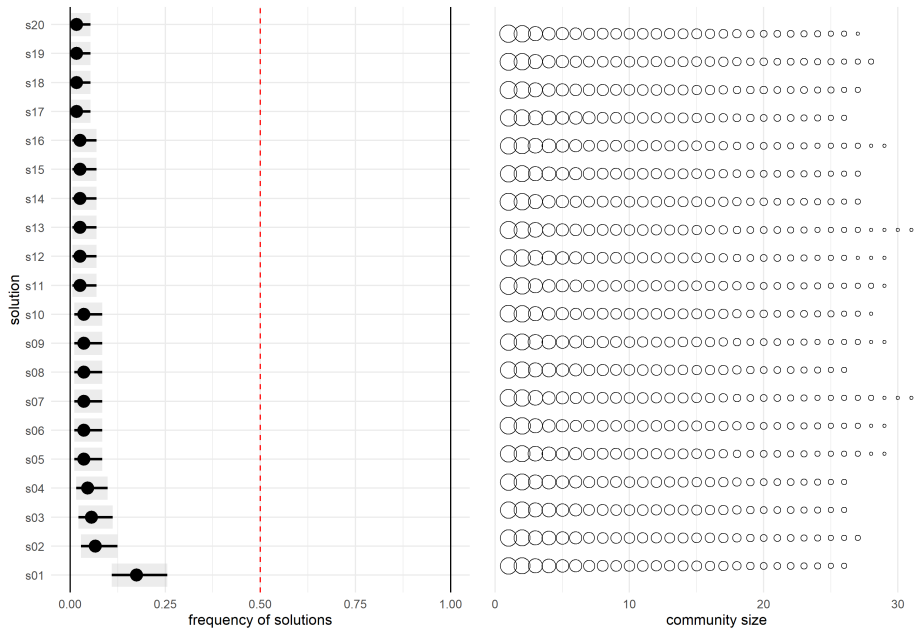


Figure 6.4: Solution space after pruning with a threshold of $w < 1.0$, displaying the top 20 communities. Despite pruning, 51 solutions remain, with the most frequent one occurring in a probability range of 0.1 to 0.25.

followed by the removal of isolated nodes, to obtain a more treatable problem.

The figure displays the solution space after pruning with a threshold of $w < 1.0$. To enhance clarity, only the top 20 communities are shown. Despite pruning, the solution space remains multiple, now with 51 solutions. Some solutions repeat, with the most frequent one occurring with a probability interval of 0.1 to 0.25.

According to the taxonomy expressed in 4 this is a "multiple" solution space. It can be reduced to a single solution using a consensus procedure, a method implemented by the communities library, as detailed in Chapter 5.

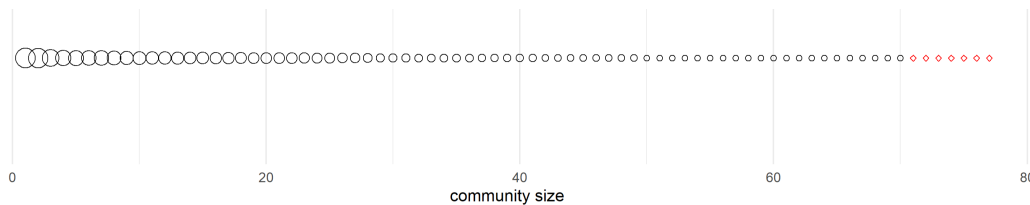


Figure 6.5: Consensus procedure applied to reduce the multiple solution space into a single partition, which is composed of 70 communities, and 7 singletons.

As a result, the final solution is shown in Figure 6.5: a single partition of 70 communities and 7 singletons, that reflects the underlying community structure

within the main component of the labor market network.

A different view of the community structure of the Labour Market Network is shown in Figure 6.6, with two options. On the left, all communities and singletons are displayed, corresponding to the "Highlight" option for managing outliers. On the right, following the "Group" option for managing outlier setting a threshold of 3 nodes. This reduces the number of relevant communities to 42, simplifying the results and improving interpretability.

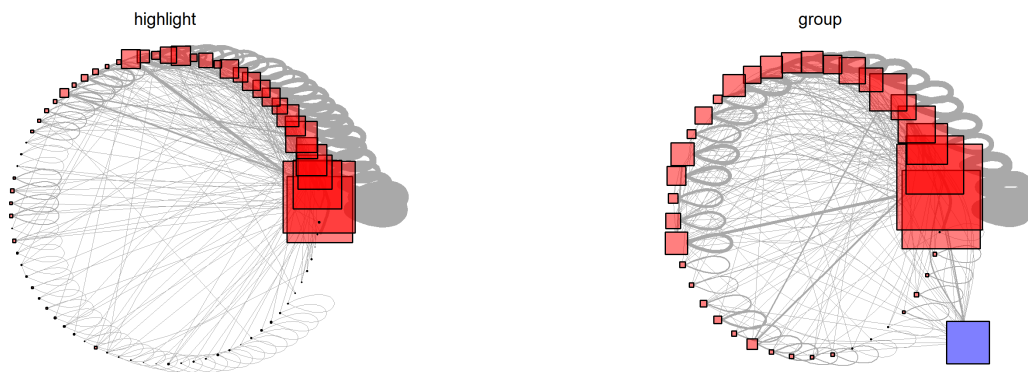


Figure 6.6: Comparison of community structure in the Labour Market Network. On the left, all communities and singletons are shown with the 'Highlight' option for outlier management. On the right, the 'Group' option is applied with a threshold of 3 nodes, reducing the number of relevant communities to 42 and simplifying the visualization. The blue square represents the group of singletons and small communities (threshold set at 3 nodes).

6.3.d Uncertainty coefficients

Consensus analysis derives an "uncertainty coefficient" in the range $(0, 1)$ for each node, as described in Chapter 4. A coefficient of 0 indicates that the node has consistently been assigned to the same community across all trials. Conversely, a coefficient of 1 indicates that the node changes community with every iteration. This measure is critical for interpretation.

A low uncertainty coefficient implies a strong and stable community assignment, suggesting that the node is firmly associated with its community. In contrast, a high uncertainty coefficient reflects a more ambiguous assignment, which may indicate that the node participates in multiple communities. This is especially important for understanding knowledge transfer, as high uncertainty suggests the node's role in facilitating knowledge flow between different groups. In such cases, the node acts as a bridge, enhancing connections across various communities.

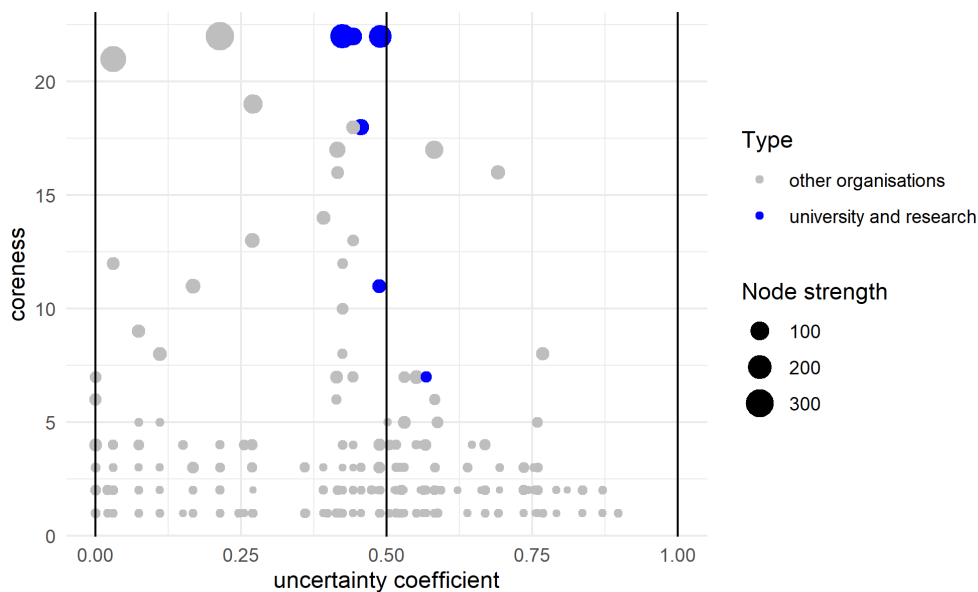


Figure 6.7: Uncertainty coefficient values for each node, indicating the stability of community assignments across multiple trials. Lower coefficients reflect consistent community membership, while higher values suggest nodes that act as bridges between multiple communities.

6.3.e Analysis of communities

Communities consist of vertices (i.e. employers) with stronger links to each other than to other communities. In terms of innovation patterns, this can be interpreted as knowledge transfer being more relevant among members of the same community than from one community to another. The fact that research centres are at the heart of their respective communities shows that the transfer of staff is an effective means of transferring knowledge, experience and innovation between academia and industry. Applying the above methodology to in Friuli Venezia Giulia region, we observed that communities are generally characterized by a central vertex (a large company, university or research center), a few prominent elements with a high proportion of membership and a large number of smaller companies.

At this point, the analysis can focus on discussing each individual community.

Below are the key details of the four main communities:

Community 1 This community consists of highly interconnected research institutions, including Università degli Studi di Trieste, Università degli Studi di Udine, Istituto Nazionale di Oceanografia e di Geofisica Sperimentale (OGS), and Istituto Nazionale di Fisica Nucleare. In addition to these academic entities, there are significant industrial players, such as Cimolai S.p.A., Calzavara S.p.A., Rhoss S.p.A., and the innovative startup EnecoLab S.r.l., which focuses on research and development activities in natural sciences and engineering. The role of this community within the network (members with high coreness) demonstrates a strong capacity for knowledge exchange, positioning it as a key hub for innovation.

Community 2 Primarily composed of industrial companies in the maritime sector, this community exhibits high centrality and plays a crucial role in specialized knowledge transfer. It is led by two major industrial players, Fincantieri S.p.A. and Fincantieri Oil & Gas S.p.A. Other key companies include RINA Services S.p.A., Monte Carlo Yachts S.p.A., and CETENA S.p.A. (Centro per gli Studi di Tecnica Navale), all of which contribute significantly to the community's influence within the maritime industry.

Community 3 This community represents another key innovation hub for the regional economy, particularly in the mechanical and automation industries. It includes large industrial companies such as Danieli & C. Officine Meccaniche S.p.A., Friul Intagli Industries S.p.A., and Modulblok S.p.A. Additionally, CAFC S.p.A. (Acquedotto Friuli Centrale), which is involved in digitalization and large-scale infrastructure investments, plays a significant role. Other important members of this community include S.P. Automation S.R.L., Fluidodinamica S.R.L., and Acciaieria Arvedi S.p.A.

Community 4 This community is heavily research-oriented, with a strong presence of prominent research institutions. These include Elettra-Sincrotrone Trieste SCPA, SISSA (Scuola Internazionale Superiore di Studi Avanzati), Istituto Officina dei Materiali - Consiglio Nazionale delle Ricerche, CERIC-ERIC, and Area Science Park. Alongside these research institutions, there is also an industrial presence, most notably Wärtsilä Italia S.p.A. and several SMEs.

As highlighted in Figure 6.9 research institutions play a prominent role in the regional labour market, as expressed by the high coreness values and their role within their community. Specifically, the two universities operating in the region (University of Trieste and University of Udine) belong to the largest community (labelled as Community 1, size 89), have comparable values of coreness and largely

surpass other large enterprises. Other major research institutions (namely Elettra Sincrotrone Trieste and the National Institute of Oceanography and Applied Geophysics - OGS) belong to the same community as the universities, with comparable strength and significantly lower values of coreness, possibly due to their sectoral specialization.

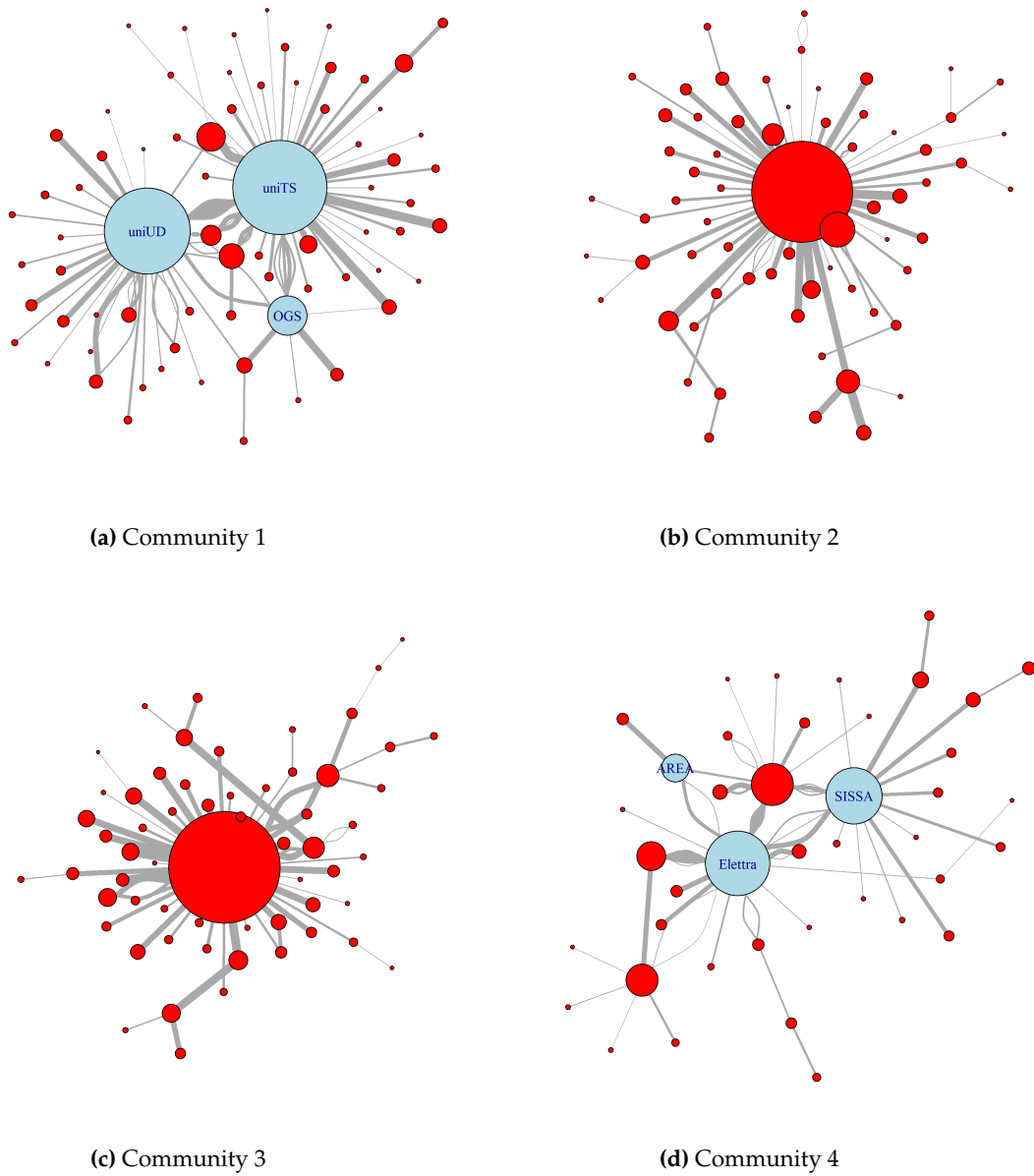


Figure 6.8: Some examples of communities. The size of vertices is proportional to their strength; colors reflect the type of organization (research organizations are in light blue, others in red). Most communities have one or a few central nodes of high cohesiveness and strength.

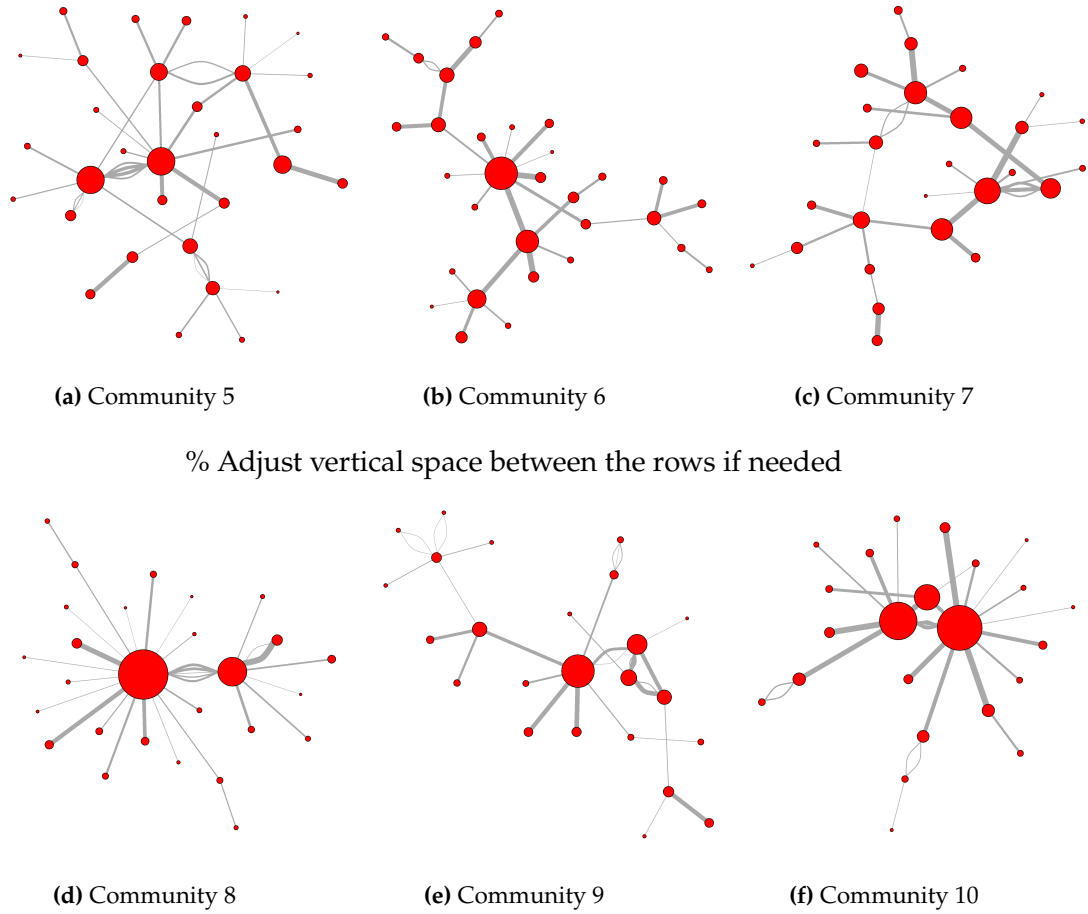


Figure 6.9: Further examples of communities. The size of vertices is proportional to their strength.

Mapping leadership and communities in EU-funded research

A significant portion of this Chapter is adapted from the paper "Mapping leadership and communities in EU-funded research through network analysis" [60], currently under peer review by Open Research Europe, the open access publishing platform for Horizon 2020 and Horizon Europe-funded research.

7.1 Horizon programmes

Horizon 2020 and Horizon Europe are flagship programs of the European Union designed to support research and innovation while fostering collaboration among companies, academic institutions, and research organizations.

Horizon 2020, launched in 2014, built on the legacy of previous European Framework Programmes such as FP7, FP6, and earlier initiatives dating back to 1984. Horizon Europe succeeds Horizon 2020, covering the period from 2021 to 2027, and introduces a mission-oriented, impact-driven approach. Its focus is on addressing major global challenges such as adaptation to climate change, cancer research, health, climate-neutral and smart cities, oceans, seas, and inland waters.

The program is mission-oriented, i.e. it aims to generate research outcomes that translate into tangible societal benefits and inform policy-making, with specific and measurable goals to be achieved by 2030.

Horizon funding is made available through open calls published on the "Funding and Tenders Portal" [32]. Calls are organized around three pillars: excellent science, global challenges and European industrial competitiveness and innovative Europe, and covers a wide range of Technology Readiness Levels (TRLs) [37], from early-stage research to close-to-market innovation. Project proposals are submitted by consortia of organizations and must address the specific challenges and expected

outcomes defined in each call. The selection process is highly competitive, with success rates remaining low due to the rigorous evaluation criteria. Nonetheless, participation remains substantial: in 2023, over 43,000 organizations participated in Horizon-funded consortia, contributing to projects with a total value of approximately €20.9 billion [34].

Projects encompass a broad range of disciplines, from natural sciences to engineering, medicine, and social sciences and humanities (SSH). Recently, these disciplines have been classified using the EuroSciVoc taxonomy [77].

A novel feature of Horizon Europe is the integration of Open Science practices to enhance transparency, reproducibility, and societal engagement, following a "data as open as possible, as closed as necessary" approach to research data. This is further discussed in Appendix A

All Horizon programmes are thoroughly documented, with open access to project data provided via the data portal data.europa.eu, ensuring wide dissemination of research outcomes.

Horizon programmes are specifically **designed to create long-term collaboration and foster innovation** among companies, academic institutions, and research organizations across multiple countries. Consortia delivering Horizon projects generally consist of at least three organizations, although larger consortia involving a dozen or more are common. The composition of these groups varies according to the specific requirements of the project and sector, often including research institutions, universities, industry partners, and small and medium-sized enterprises (SMEs). Project durations typically range between three and five years.

Although Horizon projects aim to establish these collaborations, the complexity of the resulting interactions remains challenging to capture using traditional observational methods. For instance, the approach used in [5] applies a survey-based method to analyze such collaborations, offering subjective insights into the perceived effectiveness of these partnerships.

In contrast, this thesis proposes a complementary **data-driven methodology to investigate collaboration patterns**. By utilizing data from the CORDIS database and applying advanced community detection algorithms, the proposed approach provides a scalable and objective framework for analyzing organizational interactions, community formation, and leadership roles.

7.2 The 'hydrogen energy' sector and NAHV project

A "Hydrogen Valley" is a geographical area where several hydrogen applications are combined into an integrated ecosystem that produces, exchanges, and consumes a significant amount of hydrogen, covering the entire hydrogen value chain:

production, storage, distribution, and final use [70].

Currently, over 80 hydrogen valleys are either operational or in development across Europe and beyond [91], with most in Europe funded by the Clean Hydrogen Partnership [69]. However, because these initiatives are relatively new [53], few studies have been conducted to evaluate their long-term impact on regions or track their development over time. In 2023, a Polish research team addressed this gap by applying bibliographic methods and quantitative analysis to examine the existing literature on hydrogen valleys [43]. They identified 284 publications on the topic, showing a steady increase in the number of citations. A deeper analysis revealed that the most frequently cited works focus on technological solutions and concepts for hydrogen valley development. This research highlights a significant gap in the study of how hydrogen valleys are created and evolve. Other studies, such as [72], have concentrated on the techno-economic assessment of green hydrogen valleys, examining their impacts on multiple end-users and their life-cycle assessments (LCA) [20].

The **North Adriatic Hydrogen Valley** or NAHV project [35] is one of the first transnational hydrogen valleys developed in Europe. It embraces the EU territories of Croatia, Slovenia and the Italian region Friuli Venezia Giulia and involves 37 partners based mainly on those countries. The project is co-financed by Horizon Europe programme and supported by the Clean Hydrogen Partnership [27].

Hydrogen valleys operate under two main innovation paradigms introduced in Chapter 1. The first is the open innovation model, which encourages knowledge exchange among different stakeholders involved in the valley [16]. The second is the Triple Helix model (academia, industry, government) expanded to Quintuple Helix by incorporating civil society and environmental concerns into the innovation process [12, 13].

Innovation, in the context of hydrogen valleys, is inherently driven by collaboration, particularly through partnerships between industry and academia, which combine diverse approaches and cover the entire Technology Readiness Level (TRL) scale. However, a quantitative assessment of collaborations remains elusive. The methodology proposed in this thesis seeks to address this issue by providing a methodology to identify and gauge collaboration, offering insights into the mechanisms that support innovation and the dynamics of innovation ecosystems.

From the perspective of individual organizations, understanding the dynamics of collaborations at the regional or international level can be a good opportunity. Consider, for example, an organization involved in a new hydrogen valley project: it must not only focus on the development of its own project activities, but it may benefit from identifying partners and competitors that are engaged in other hydrogen valleys. This broader awareness may help identifying synergies, risks,

and market opportunities.

Policymakers adopt yet another perspective. They have a strategic vision to align local, national, and European resources, aiming for medium- and long-term impact of the innovation ecosystems. Achieving this requires a clear understanding of how collaborations emerge between industry and research, who the key stakeholders are, and how these collaborations evolve over time.

In both cases, two research questions are essential:

1. Which organisations are the most influential in driving research and innovation? Identifying these key players — whether companies, academic institutions, or research organisations — is important for other organisation that want to be in contact with them in future projects. Moreover, policy makers may be interested in assessing whether the policy put in place in their territory has produced an improvement of leadership roles over the years.
2. Are EU-funded projects fostering partnerships that extend beyond the duration and scope of individual projects? If such long-term communities exist, they serve as vital indicators of the open innovation and quintuple helix models effectiveness. Stable communities suggest robust exchanges of ideas and collaboration, which are essential for sustaining innovation and achieving the long-term goals of hydrogen valleys.

Key requirements of this approach are that results must be independent of contingent factors (such as software implementation or ordering of the input data) and tested for validity. Additionally, when multiple algorithmic options are available, the selection must be data-driven and performance-oriented, ensuring that the chosen algorithm yields the most interpretable and reliable outcomes.

7.3 Data acquisition

The source of data for this study is CORDIS (Community Research and Development Information Service), the European Commission’s primary public repository and portal to disseminate information on all EU-funded research projects and their outcomes [31]. It provides open access to comprehensive data on projects, including objectives, participants, funding details, and results. For the purpose of this study, two datasets have been extracted: Horizon 2020 (projects starting from 2014 to 2020) [29] and Horizon Europe (projects starting from 2021 to 2027) [30]. The two datasets have the same structure, composed of several tables, the most relevant being the organisations table denoted as **O** and the projects table denoted as **P**.

Each project is characterized by a unique identifier (`projID`), its acronym, title, start date, end date. Moreover, each project is associated with a long textual field, describing its objectives, and with structured categorical information on the call and funding scheme. A separated table associates each project with one or

more topics, encoded according to the European Science Vocabulary (*EuroSciVoc*), the multilingual taxonomy representing scientific fields developed by the EU's Publications Office.

There is a many-to-many relationship between organisations and projects (i.e. a project has several organisations, and an organisation can be part of several projects). In table \mathbf{O}^* each organisation is identified by a unique identifier (*orgID*), a name, detailed geolocation information. In addition, the table records the role of each organisation in each project (coordinator, participant, or associated partner), as well as the total costs incurred by the organisation within each project (*totalCost*) and the corresponding eligible contribution from the Horizon programme (*netEcContribution*).

Data representation can be optimized by merging the corresponding tables for Horizon 2020 and Horizon Europe, to obtain the following core data structures:

- table \mathbf{P}^* listing all projects, their *projID*, start and end dates, reference to the Horizon calls for proposal, keywords, and a long textual descriptions of project objectives;
- table \mathbf{O}^* listing all organisations, their *orgID* location, their role in the project, and monetary values *totalCost* and *netEcContribution*;
- table \mathbf{T}^* associating each project in P to one or more keywords of the *EuroSciVoc* taxonomy.

7.4 Data preparation

The initial step in data preparation involves identifying the subset of projects $\mathbf{P} \subset \mathbf{P}^*$ on which we aim to focus, based on the topics from that are relevant for the objectives of the analysis. Consequently, the other tables can be filtered to ensure that only pertinent data is retained in the form of $\mathbf{O} \in \mathbf{O}^*$ (organisation name and location). The core information for our analysis the weights table denoted as \mathbf{W} , which encodes the annual effort each organisation $o \in \mathbf{O}$ puts into each project $p \in \mathbf{P}$ in a given year. A proxy for the values in W can be derived from either *netEcContribution* or *totalCost*. The former indicates the total amount of public funding received by an organisation upon project completion and serves as a useful proxy for how much the Horizon grant promotes collaboration. The latter reflects the total cost incurred by the organisation to complete the project. For non-profit organisations involved in research projects, these values often coincide, as the Horizon grant may cover 100% of the costs. However, for private companies investing in pilot projects, such as in the NAHV project, the two values can differ

significantly, sometimes by an order of magnitude. In the case study presented in section 7.5, the weight is based on `netEcContribution`, expressed in thousands of euros.

The data is then segmented on a yearly basis. Denoting by \mathbf{W}_y the matrix for a given year y , it is a rectangular matrix of size $|\text{projID}| \times |\text{orgID}|$. The value of each project is assumed to be equally distributed over its duration, from its start date to its end date. Consequently, the contribution of a project to the matrix \mathbf{W}_y is divided proportionally across the years during which the project is active.

Textual fields in table \mathbf{P} contain valuable information, but in a format unsuitable for network analysis. The project objectives are typically described in hundreds of words, and user-defined keywords lack a standardized vocabulary. To make this information usable, the last step in data preparation involves creating Boolean attributes for each project. These attributes encode relevant information in a simplified format, such as whether the project is focused on technology or market uptake, or whether hydrogen is a primary focus or one of several applications. This was accomplished using a script that interacts with the API of a Large Language Model, which processes the lengthy textual fields and generates a dataframe, indexed with `projID` and one or more Boolean variables, which can be merged with \mathbf{P} .

7.5 Results

This section presents the results obtained applying the methodology outlined in Chapter 4 to a subset of hydrogen-related projects, including the NAHV project from 2015 to 2029. The analysis aims to investigate leadership roles, community structures, and their evolution over time. Furthermore, the analysis demonstrates how AI-generated keywords can be integrated to provide additional insights into the distinction between market-oriented and technology-oriented projects.

The data has been prepared using the *EuroSciVoc* topic *hydrogen energy*, and the AI-generated categories are market vs technology, as explained in Section 7.4. The family of networks $\{G_{2015} \dots G_{2024}\}$ has a remarkable evolution over time. G_{2015} is composed of 83 organisations, and the number steadily increases to 648 by 2024. The number of collaborations, represented by the network's edges, grew proportionally during this period, and the total value of projects, measured by the sum of weights in the network, saw an even more significant increase, rising from 547 to 6628. Looking ahead to the period from 2025 to 2029, the future of collaborations will be shaped by contracts currently in operation and influenced by upcoming calls and new projects. As shown in Figure 7.1, after 2024 the number of partners and collaborations decreases, while the total investment remains stable for the next two years.



Figure 7.1: Size of the networks G_y represented by the number of organisations (horizontal axis) and the number of collaborations (vertical axis) and total investment per year (bubble size). Between 2016 and 2024 there is an increase in the number of organisations from 83 to 648 and a proportional rise in collaborations as represented by weights, rising from 547 to 6628. Current data for projects continuing in years 2025 to 2029 suggest further growth in the total value of projects.

Leadership roles

The methodology for AI-generated categories, discussed in 7.4, is applied to the hydrogen projects dataset. This classification approach distinguishes between projects focused on market development — encompassing policy, market uptake, business models, and hydrogen valleys — and those centred on technological development. The NAHV project clearly falls into the former category. This classification enables the creation of two distinct families of networks, denoted as G_y^M and G_y^T . The total value of the projects in each group is shown in Figure 7.2, revealing an interesting trend: while investment in market-oriented projects has been consistently present since 2016, there is a notable increase in such investments starting in 2023.

Figure 7.3 illustrates how centrality measures and AI-generated categories provide insights into the roles of organisations in hydrogen-related innovations, highlighting differences between those engaged in market-oriented and technology-oriented projects.

The figure shows in grey the range of values found each year, and highlights organisations participating in the NAHV project. The diagrams provide an overview of the evolution of coreness in G_y^M (left) and G_y^T (right). The grey shaded area represents the range of coreness values observed each year. The organisations involved in the NAHV project are highlighted with coloured dots, connected by lines

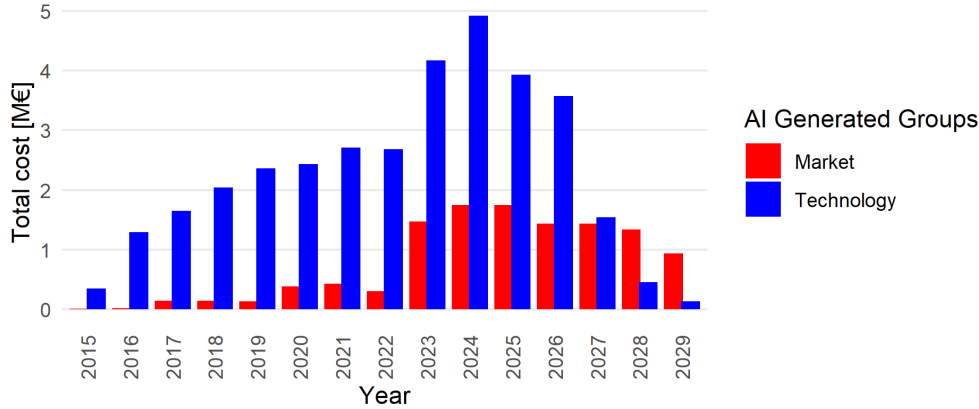


Figure 7.2: Comparison between technology-oriented and market-oriented project. The value is measured by netEcContribution; groups are identified by the AI generated labels: G_y^M and G_y^T

that illustrate their progression over time. A notable observation is that, while all NAHV partners are part of G_y^M (which is expected, given NAHV’s classification as a market project), only a few are involved in G_y^T .

Coreness values can fluctuate over time: an increase suggests that an organisation is gaining influence through participation in relevant projects, while a decrease may indicate that, following the completion of a project, the organisation has not yet initiated a new one within the Horizon framework. The analysis also includes the years 2024 to 2029, reflecting ongoing projects that are scheduled to conclude during this period. It is important to note that this is not a forecast, but rather a record of existing projects with future completion dates.

Moreover, centrality of all organisations can be visualized in a degree-coreness plane, where each organisation is represented by a bubble. Degree serves as a proxy for an organisation’s capacity to form partnerships with many other organisations, while coreness reflects its ability to partner with influential organisations. Although coreness is limited by degree, the size of the bubble represents the organisation’s strength, which serves as a proxy for its capacity to attract substantial funding and invest in large projects. An example is shown in figure 7.7, which also includes an information about communities.

Communities In the context of this paper, communities refer to relevant sub-groups of organisations that exhibit strong collaborative ties. The focus is on analysing a single partition and its evolution over time. The analysis is conducted on the family of complete networks, G_y .

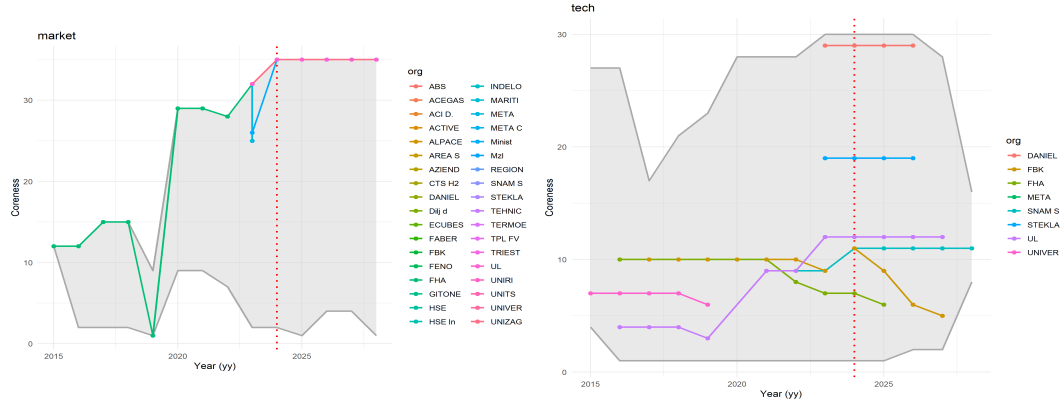


Figure 7.3: Coreness of organisations involved in the NAHV project, shown in comparison with the range of coreness values for each year (grey background).

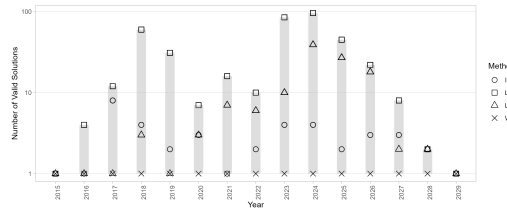


Figure 7.4: Number of solutions $|\mathcal{S}_y|$ generated by different community detection algorithms.

As explained in Chapter 4, the first step involves exploring the solution space \mathcal{S}_y^A in terms of completeness, the number of solutions, and their validity across various algorithms. The tested algorithms include Louvain (LV) [9], Leiden (LD) [90], Label Propagation (LP) [79], Edge Betweenness (EB) [65], Leading Eigenvector (EV) [64], Walktrap (WT) [74], and Infomap (IM) [83]. The tests were conducted in R [78], using the iGraph [21] and communities [57] libraries.

The number of solutions $|\mathcal{S}_y^A|$ are illustrated in 7.4: most algorithms produced multiple solutions across most years, with the exceptions of 2015 and 2029, where the networks were simpler. Interestingly, WT consistently provided a unique solution. However, EB had a different issue, generating partitions that were not similar to each other, with similarity coefficients often below 0.5. EV produced multiple valid solutions in most years but failed to find a partition in 2024.

A more in-depth observation of the solution space for 2024 (the most critical year) reveals additional details, as shown in 7.5. LV generates a large set of solutions with $|\mathcal{S}_{2024}^{LV}| = 69$, which are also highly diverse, as evidenced by the similarity plot showing a bimodal distribution with peaks around 0.6 and 0.9. Moreover, ap-

proximately one-third of the solutions are invalid due to containing one or more internally disconnected communities. This is a well-documented issue of the algorithm, as discussed in the literature. LD, specifically designed to address the problem of disconnected communities, also produced invalid solutions for this network: its partition consists of a few large communities and a high proportion of singletons, with a mixing parameter exceeding 0.5, indicating that nodes within a community were more connected to nodes outside their community than to those within it.

LP generates a different solution with each run; in this case, with $t = 100$, there are 96 valid and distinct solutions, while 4 are invalid due to internally disconnected communities. Better results can be achieved with IM that produces $|\mathcal{S}^{IM}| = 1$ in most years, and a dominant one solution in the most complex cases such as 2024. Overall, WT is the only algorithm that consistently provided valid results across all years, with valid partitions and $|\mathcal{S}^{WT}| = 1$, consequently, it will be used for community detection and temporal evolution analysis.

The selected partition \mathcal{P}_y can be represented as a network in which nodes belonging to the same community are grouped and coloured to distinguish between communities. For example, the result for \mathcal{P}_{2018} is shown in Figure 7.6. Some communities appear disconnected from the others; these consist of organisations that, in that year, participated in only one project and thus form a separate component.

Community detection can be integrated with centrality measures to provide a more comprehensive understanding of network structure.

Figure 7.7 illustrates the results for \mathcal{P}_{2018} . In this visualisation, each bubble represents an organisation, with its position determined by coreness and degree centrality. Bubble colours correspond to the community, and bubble size is proportional to the strength (the total weight of its connections in the given year). The diagram clearly demonstrates that degree is the upper limit for coreness, as no points appear above the line with a slope of 1. In practical terms, this indicates that an organisation that participates in many projects (high degree) is not necessarily the most influential in the network. Strength, which reflects the total weight of an organisation's connections, remains independent of both degree and coreness, meaning that large bubbles can appear anywhere on the graph. This means that some organisations, despite having relatively few connections (low degree), are linked to relevant partners (high coreness) and stand out for their ability to secure significant EU research funding (high strength).

Evolution of communities Temporal evolution was calculated as described in Chapter 4, using the WT algorithm to ensure the robustness of the results. In the diagram shown in Figure 7.8, the horizontal axis represents time, with each community depicted as a coloured rectangle. The height of each rectangle is pro-

portional to the sum of the strengths of its members. For the purpose of clarity, only the six largest communities are represented. The analysis reveals a significant pattern: initially a large number of small communities is formed, and their configuration changes rapidly. This reflects the fact that in Horizon 2020 public funding was primarily allocated to technology-oriented research projects, and partnerships changed rapidly. However, with the onset of Horizon Europe, and specifically the Clean Hydrogen partnership, three large communities emerge, and are set to continued collaboration through 2029.

The significant size of the largest communities reflects the increasing allocation of resources to Horizon Europe projects in the hydrogen sector. This growth is driven by the development of hydrogen valleys, which have fostered long-term, stable partnerships, contributing to both the stability of collaborations and the continued expansion of these communities.

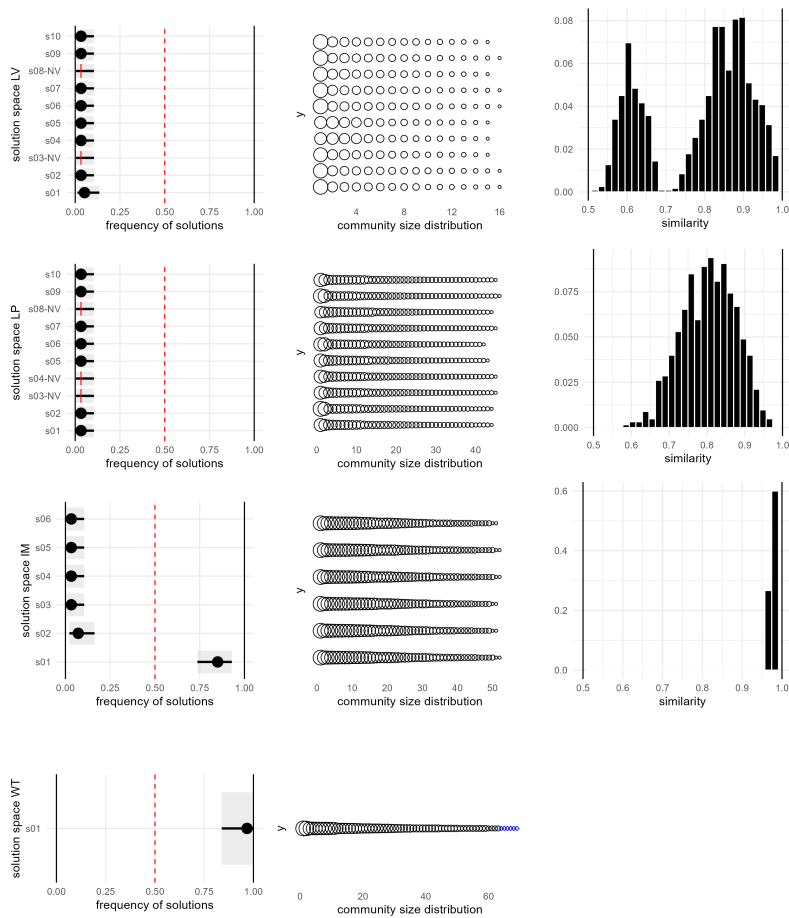


Figure 7.5: Solution space diagrams. Left: frequency and confidence intervals for the solutions identified by each algorithm. In the case of LV and LP only the 10 most frequent solutions are shown. Middle: distribution of community size for each solution. Proper communities are represented as black circles, single-node communities are represented as blue diamonds. Right: pairwise similarity between solutions.

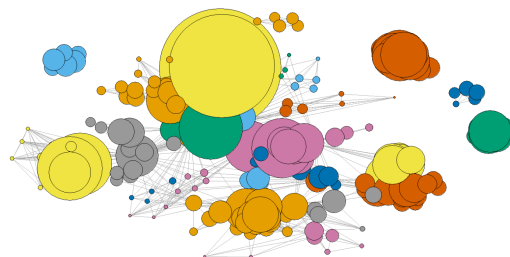


Figure 7.6: Network G_{2018} , with communities grouped and highlighted in different colours.

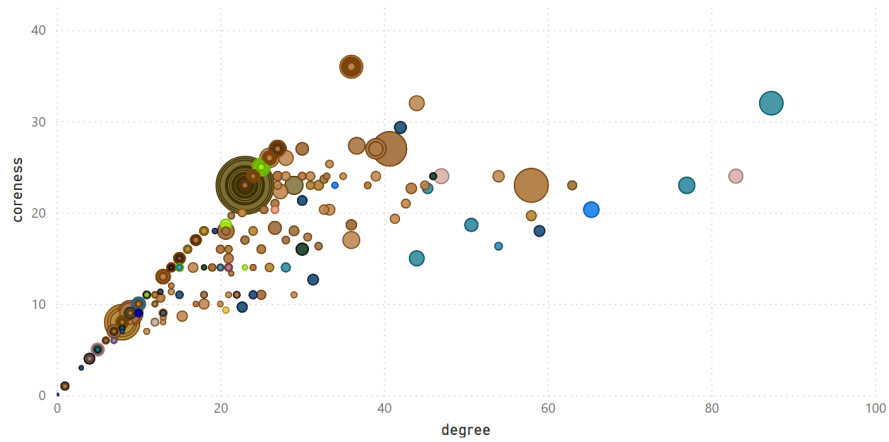


Figure 7.7: Centrality measures of all organisations in G_{2024} . Each bubble represents an organisation in the degree-coreness plane. Degree is a proxy for the organisation’s ability to win projects and attract funding, while coreness reflects its capacity to partner with other influential organisations. The largest bubbles in the top-right of the diagram highlight organisations that excel in both areas.

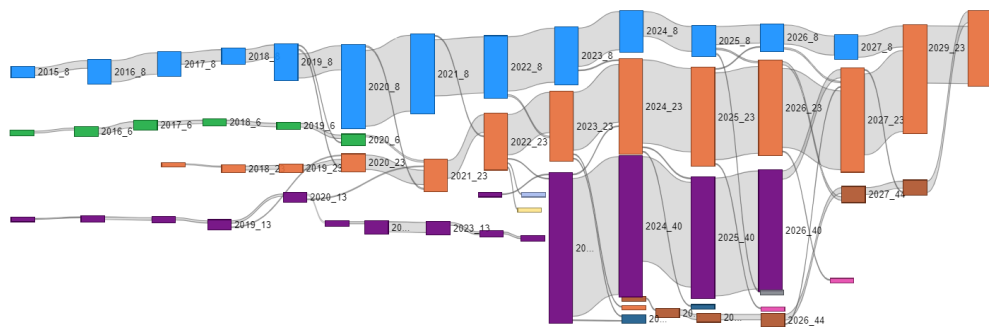


Figure 7.8: Evolution of the largest communities from 2015 to 2029. For clarity, only the six largest communities are displayed.

Conclusions

The research presented in this thesis contributes to the advancement of community detection in complex networks, specifically addressing the challenges of stability and reproducibility. Through the development and application of a novel methodology, based on the concepts of *solution space* and *consensus*, this work moves beyond the traditional single-solution approach. Moreover, the proposed consensus methodology introduces a taxonomy for outliers (incorporate, highlight or group) and an *uncertainty coefficient* associated with each node of the network, which captures the inherently fuzzy nature of community structures in complex networks and offers a more nuanced view of the network.

To validate this paradigm, a series of tests were conducted on artificial networks, using benchmarks like the RC and LFR models. These experiments demonstrated the effectiveness of the proposed workflow in overcoming limitations such as input-order bias, variability of solutions, and the presence of outliers.

Aligned with open science principles, the full code and datasets underlying this research are disclosed, alongside the development of an R library, *communities*, which provides tools for the generation of benchmark networks, solution space exploration, and consensus community detection.

The research is grounded in the practical question of demonstrating how network analysis can provide useful insights into *collaborations* that lead to *innovation*, specifically identifying leading organisations, cohesive groups and their evolution over time.

Two case studies were used as practical demonstrations of this approach. The first case study, focused on the labour market network in Friuli Venezia Giulia, demonstrated how regional collaboration networks between firms, universities, and research centres can reveal key players in innovation and track their evolving roles over time. Through network analysis, we identified which organizations were most

central to fostering innovation and how communities of collaborators formed and dissolved, reflecting the dynamic nature of regional innovation ecosystems. The second case study applied the same methodology to the Horizon-funded hydrogen energy sector. Here, the analysis revealed the collaborative structures within EU-funded research projects, illustrating how partnerships between industries and research institutions extended beyond individual projects to form lasting innovation communities. This finding supports the open innovation model discussed in Chapter 1, where external collaborations between various actors drive technological advancements and the diffusion of knowledge.

Policy Recommendations

From an applied perspective, the methodology presented in this thesis offers valuable tools for policymakers, especially those involved in the Hydrogen Valley project, as discussed in Chapter 7. Policymakers across regions such as Friuli Venezia Giulia, Croatia, and Slovenia can use this approach to monitor the engagement and collaboration of organizations within their territories. As the initial Hydrogen Valley projects (launched in 2022-2023) conclude and new Horizon calls become available, policymakers will be equipped to assess whether today's leading organizations are forming new partnerships and maintaining their leadership roles. Additionally, at the European level, this methodology can offer insights into the evolution of research and innovation communities, helping policymakers gauge the openness of these communities to newcomers and the stability of collaboration between industrial and academic partners.

Moreover, institutions such as Area Science Park can apply the methodology to further case studies on diverse topics, particularly pandemic preparedness and electron microscopy—two areas of interest for Area Science Park that have been explored in the published dataset Horizon Projects Network 2024 [58].

Finally, policymakers should consider enhancing their evaluation frameworks by linking the results of network analysis proposed in this thesis, with the detailed textual information on project deliverables available on the CORDIS platform. Deliverables are linked to projects and organizations, hence to a temporal and geographical framework; each provides valuable insights into ongoing innovations carried out by local companies or research centres, as well as emerging innovation trends in competing regions. This approach supports evidence-based decision-making, helps identify synergies, prioritize funding opportunities, and update regional Smart Specialisation Strategies (S3).

Methodological Developments

This thesis opens the door to further research in several areas, including further testing and validation of the methodology for solution space exploration and consensus, enhanced software implementation, and its application to diverse domains.

Expanding tests to larger, denser networks, with artificial or real-world topologies, will provide a comprehensive evaluation of algorithm performance and may reveal potential limitations or challenges. The inclusion of additional algorithms would further enhance the understanding of the robustness of the methodology. While this work focuses on widely used methods such as LV, LD, IM, LP, and WT, incorporating approaches based on Edge Betweenness and Eigenvector methods could offer new perspectives on community structures.

A further extension of the testing and benchmarking set should be focused on the capacity (or tendency) of each algorithm to generate a specific type of solution space. This can be done initially on simulating controlled benchmark networks, such as a *Ring of Cliques* with varying parameters (e.g., clique numbers n_c ranging from 2 to 20 and clique sizes c_s ranging from 3 to 30), while maintaining consistent random seeds, will support systematic comparisons and performance evaluations.

Testing the proposed methodology on complex synthetic networks is essential to validate its robustness and applicability to real-world systems. LFR networks are particularly suitable as they generate realistic community structures by incorporating overlapping communities and power-law distributions of node degree and community size. By varying the network size (e.g., $n = 1000$ or larger), would allow to evaluate the methodology's scalability and performance under increasing levels of complexity and heterogeneity.

Additional simulations using models that closely mimic real-world interactions, yet remain under controlled conditions, should also be tested. Examples include Small-World networks, characterized by high clustering and short path lengths, which reflect systems where localized connections coexist with long-range links (e.g., social or biological networks), and Preferential Attachment models, which produce scale-free networks where highly connected nodes attract more links over time, mirroring growth dynamics in systems such as social networks or the internet.

The proposed methods, which primarily rely on R and the iGraph library, are flexible but notoriously slow. Exploring the solutions spaces scales the computational effort linearly with the maximum number of iterations, which can typically reach stability after about 50 iteration for a Single solution, and may well exceed 300 for a Sparse solution space. While this overload may be negligible in controlled test settings, it could pose severe challenges when applied to large networks, and raises an obvious question of performance, which must be thoroughly tested and optimized.

Software implementation can help with performance: re-implementing key components in high-performance languages such as C++ could substantially reduce execution time and improve scalability, enabling applications to large-scale networks.

To extend accessibility, developing a Python library equivalent to the current R implementation would cater to a broader user base in data science, network analysis, and machine learning. Furthermore, it should be considered to develop a version compatible with other network analysis libraries, notably NetworkX for R and for Python to further facilitate integration into existing workflows.

Finally, the methodology proposed in this thesis can also be applied to various domains, such as social networks (to uncover patterns of influence, opinion dynamics, and organizational behaviour), citation networks (to identify influential publications, collaborative trends, and emerging research communities), and mobility networks (to detect geographical or temporal patterns as communities). Exploring these and similar domains will further demonstrate the method's utility for both academic research and practical applications.

Appendix: Open Science

Open Science is an approach to scientific research that emphasizes transparency, accessibility, and collaboration. Its main objective is to make the process and results of research widely available, to ensure that scientific findings can be reproduced and validated by the global research community. It encompasses several key practices: FAIR data management, open access to publications, data and software. Open Science also encourages early sharing of results, including preprints and data in open repositories, which accelerates the pace of discovery and fosters greater collaboration among scientists, policymakers, and the public. The European Commission strongly advocates for Open Science practices, integrating them into its major funding programs, such as Horizon Europe.

In this thesis, Open Science principles have been embraced, with particular focus on the open access to data and code.

A.1 FAIR principles

The FAIR principles (Findable, Accessible, Interoperable, and Reusable) provide a framework for ensuring data management practices align with open science principles [92, 55]. FAIR principles guide researchers in making data more discoverable, accessible, and reusable by both humans and machines. Below is a summary of the key aspects of each principle:


Findable Data must be easily findable by both humans and machines. This is crucial for ensuring that datasets can be located through standard search mechanisms. Findability can be achieved through persistent identifiers (such as DOIs), rich metadata that describe the data, and all necessary details about its content, origin, and conditions for reuse. Moreover, datasets should be


registered in searchable repositories so they can be discovered by common search engines or specialized research portals.

Accessible Once the data is found, it must be accessible, meaning that users should be able to retrieve it and clearly understand the conditions under which it can be accessed and utilized. Access protocols should be standardized and transparent. Authentication protocols should ensure that data access is controlled and secure. However, some types of data may justifiably remain inaccessible, such as personal information, medical records, or trade secrets. Additionally, certain data may be temporarily embargoed, as in the case of patents. This principle is often summarized as: "As open as possible, as closed as necessary." Even when data is restricted, the corresponding metadata should always remain available.

Interoperable Data should be interoperable with other datasets and systems, so it can be integrated with other research outputs. Standardized formats and vocabularies should ensure consistency across different fields of research.

Reusable Open science advocates for making data reusable to replicate scientific results and enable the reuse of data for different research purposes. Clear licenses must define the terms of reuse, including any restrictions or conditions.

 The data used and generated in this thesis complies with the FAIR principles, ensuring it is findable, accessible, interoperable, and reusable. It has been published in Zenodo [66], the data repository funded by the European Commission through the OpenAIRE project. Zenodo is fully compliant to FAIR principles, providing an open-access platform to store, share, and publish datasets, software, and research outputs.

 Findability is further enhanced by using persistent identifiers that connect datasets (via DOIs) to individual researchers. ORCID (Open Researcher and Contributor ID) provides a unique and permanent identifier for researchers, ensuring accurate attribution of their work across various platforms. The Author's ORCID iD is 0000-0002-2034-2951.

A.2 Open publication

Open Publication refers to the practice of making research outputs freely accessible to everyone, eliminating barriers such as paywalls or subscription fees. This approach ensures that research results can be reviewed, replicated, and expanded

upon by a wide and diverse audience, including other researchers, policymakers, and the general public.

Open Research Europe (ORE) [36] is an open publication platform dedicated exclusively to the dissemination of research results funded by the European Commission. Key features of ORE include free access for authors, full compliance with FAIR principles, rapid publication, and a transparent open peer review process. Manuscripts are made publicly available shortly after submission, with the peer review process commencing post-publication. The identities of reviewers and their reports are published alongside the articles, promoting accountability and providing readers with further valuable information.

A.3 Open access to data

Open access to data is a core Open Science practice, according to FAIR principles. This section provides information on open access to both the underlying data and the data generated throughout this thesis.

Underlying Data

The underlying data for this research is openly available through data.europa.eu, the official portal for European data. Specifically, data are accessible from the following links:

Horizon 2020 <https://data.europa.eu/data/datasets/cordish2020projects>

Horizon Europe <https://data.europa.eu/data/datasets/cordis-eu-research-projects-under-horizon-europe-2021-2027>

Extended Data

The data generated for this thesis area available under the terms of the Creative Commons Attribution 4.0 International license (CC-BY 4.0), in Zenodo. Developed by CERN under the European OpenAIRE program, Zenodo is an open-access repository for sharing and preserving research outputs. It is *FAIR-by-design*, ensuring that data are Findable, Accessible, Interoperable, and Reusable through comprehensive metadata, persistent identifiers and open access policies.

Repository: Horizon Projects Network

DOI: <https://doi.org/10.5281/zenodo.13765372>


The project contains 3 case studies, each focused on a specific topic, selected using the EuroSciVoc taxonomy: hydrogen energy (analysed in chapter 7), electron microscopy, and pandemics. Each case study contains the following files:


- **O.csv**: Contains organisations' unique identifiers and attributes.
- **P.csv**: Contains projects' unique identifiers and attributes.
- **W.csv**: Describes the participation of each organization in a project in each specific year. The participation is weighted according to the "total cost" of the project, shared proportionally based on the project's duration within the year.
- **activity_type.csv** provides a description of the codes used to define the activity types associated with each organization.
- **2015.graphml, 2016.graphml, ..., 2029.graphml**: a set of files representing network model for collaboration between organizations in any given year from 2015 to 2029. Each graph encodes the centrality measures (degree, strength, coreness) and community labels.
- **format-info.txt** provides information on the format of **.csv** and **.graphml** files used in this dataset.
- **sample-networks-hydrogen-energy.pdf** showcases a visual representation of the networks from 2015 to 2029.


GraphML (Graph Markup Language) is an open XML-based format specifically designed to describe networks. It is based on XML (eXtensible Markup Language), a text-based machine-readable format. GraphML format supports directed, undirected, and mixed graphs, node and edge attributes as well as hypergraphs. A detailed description of the GraphML format is available in the official GraphML website [19] and documented for example by [10].

A.4 Open access to code and software

All software used for this thesis is available under the principles of open science.

 The code for this project has been developed using the R programming language [78], a widely-used open-source language designed for statistical computing and data analysis, that provides a powerful environment for manipulating data, conducting statistical analyses, and producing high-quality visualizations.

 **igraph** Network analysis in this project is done with the R-igraph package [21], that offers a comprehensive set of functions for creating, manipulating, and visualizing networks.

 The development version of the code used for this project is available through GitHub <https://www.github.org> an open platform for hosting, sharing, and collaborating on software development projects built on the Git distributed version control system. development. The following repositories are available:

CCD package The *CCD* package, developed as part of this research, to implement the first version of consensus community detection as published in [59] and [62].

Available at: <https://github.com/fabio-morea/CCD>

Communities package The *communities* package [57], developed as part of this research, to implement the final version of solution space exploration and consensus community detection.

Available at <https://github.com/fabio-morea/communities>

Horizon intelligence code for data preparation, enrichment and network analysis on the Horizon case study in Chapter 7.

Available at <https://github.com/fabio-morea/horizon-intelligence>.



Acknowledgments

The author gratefully acknowledges the Regional Observatory on Policies and the Labour Market of the Friuli Venezia Giulia Region for providing the data used in the Labour Market network analysis presented in Chapter 5.

The case study on Hydrogen Energy has been developed as part of the North Adriatic Hydrogen Valley (NAHV) project. The project received support from the European Union under grant agreement 101111927. The views and opinions expressed in this paper are solely those of the author(s) and do not necessarily reflect the official positions of the European Union or the Clean Hydrogen Joint Undertaking. Neither the European Union nor the granting authority assumes responsibility for the content presented.

Bibliography

- [1] Edo Airoldi et al. “Mixed membership stochastic blockmodels”. In: *Advances in neural information processing systems* 21 (2008).
- [2] ARWA Aldabobi, AHMAD Sharieh, and RIAD Jabri. “An improved Louvain algorithm based on Node importance for Community detection”. In: *Journal of Theoretical and Applied Information Technology* 100.23 (2022), pp. 1–14.
- [3] Vladimir Batagelj and Matjaz Zaversnik. “An O(m) Algorithm for Cores Decomposition of Networks”. In: *ArXiv cs.DS/0310049* (2003). URL: <https://api.semanticscholar.org/CorpusID:15799869>.
- [4] Stefano Benati et al. “Overlapping communities detection through weighted graph community games”. In: *PLOS ONE* 18.4 (Apr. 2023), pp. 1–35. DOI: 10.1371/journal.pone.0283857. URL: <https://doi.org/10.1371/journal.pone.0283857>.
- [5] Takwa Benissa and Anish Patil. “Drivers for Clustering and Inter-Project Collaboration—A Case of Horizon Europe Projects”. In: *Administrative Sciences* 14.5 (2024). ISSN: 2076-3387. DOI: 10.3390/admsci14050104. URL: <https://www.mdpi.com/2076-3387/14/5/104>.
- [6] Alberto Bertello, Paola De Bernardi, and Francesca Ricciardi. “Open innovation: status quo and quo vadis - an analysis of a research field”. In: *Review of Managerial Science* 18.2 (2024), pp. 633–683. ISSN: 1863-6691. DOI: 10.1007/s11846-023-00655-8. URL: <https://doi.org/10.1007/s11846-023-00655-8>.
- [7] M. Bjelland et al. “Employer-to-employer flows in the United States: estimates using linked employer-employee data”. In: *Journal of Business and Economic Statistics* 29.4 (2011), pp. 493–505.

- [8] Vincent Blondel, Jean-Loup Guillaume, and Renaud Lambiotte. "Fast unfolding of communities in large networks: 15 years later". In: *Journal of Statistical Mechanics: Theory and Experiment* 2024.10 (Oct. 2024), 10R001. DOI: 10.1088/1742-5468/ad6139. URL: <https://dx.doi.org/10.1088/1742-5468/ad6139>.
- [9] Vincent D Blondel et al. "Fast unfolding of communities in large networks". In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008.
- [10] Ulrik Brandes et al. "Graph Markup Language (GraphML)". In: *Handbook of Graph Drawing and Visualization*. Ed. by Roberto Tamassia. Boca Raton, FL: CRC Press, 2013, pp. 517–541. URL: <https://www.uni-konstanz.de/mmisp/pubsys/publishedFiles/BrEiLe10.pdf>.
- [11] Matthew Burgess, Eytan Adar, and Michael Cafarella. "Link-prediction enhanced consensus clustering for complex networks". In: *PloS one* 11.5 (2016), e0153384.
- [12] Elias G. Carayannis and David F.J. Campbell. "Triple Helix, Quadruple Helix and Quintuple Helix and how knowledge, innovation, and the environment relate to each other: A proposed framework for a trans-disciplinary analysis of sustainable development and social ecology". In: *International Journal of Social Ecology and Sustainable Development (IJSESD)* 1.1 (2010), pp. 41–69. DOI: 10.4018/978-1-4666-0882-5.ch3.8.
- [13] Andreana Casaramona, Antonia Sapia, and Alberto Soraci. "How TOI and the quadruple and quintuple helix innovation system can support the development of a new model of international cooperation". In: *Journal of the Knowledge Economy* 6.3 (2015), pp. 505–521.
- [14] Tanmoy Chakraborty et al. "Constant communities in complex networks". In: *Scientific reports* 3.1 (2013), p. 1825.
- [15] Tanmoy Chakraborty et al. "Ensemble Detection and Analysis of Communities in Complex Networks". In: 1.1 (Mar. 2020). ISSN: 2691-1922. DOI: 10.1145/3313374. URL: <https://doi.org/10.1145/3313374>.
- [16] Henry Chesbrough and Marcel Bogers. "3Explicating Open Innovation: Clarifying an Emerging Paradigm for Understanding Innovation". In: *New Frontiers in Open Innovation*. Oxford University Press, Nov. 2014. ISBN: 9780199682461. DOI: 10.1093/acprof:oso/9780199682461.003.0001. eprint: https://academic.oup.com/book/0/chapter/148736254/chapter-ag-pdf/44989927/book/_5676/_section/_148736254.ag.pdf. URL: <https://doi.org/10.1093/acprof:oso/9780199682461.003.0001>.

- [17] Julien Chiquet et al. *R package aricode: Efficient Computations of Standard Clustering Comparison Measures*. R package version 1.0.1. 2022. URL: <https://CRAN.R-project.org/package=aricode>.
- [18] Aaron Clauset, Christopher Moore, and Mark EJ Newman. “Hierarchical structure and the prediction of missing links in networks”. In: *Nature* 453.7191 (2008), pp. 98–101.
- [19] Graph Drawing Community. *GraphML: The Graph Markup Language*. Accessed: 2024-9-20. 2021. URL: <http://graphml.graphdrawing.org/>.
- [20] Giulia Concas et al. “Life Cycle Analysis of a Hydrogen Valley with Multiple End Users”. In: *Journal of Physics: Conference Series* 2385 (2022), p. 012035. DOI: [10.1088/1742-6596/2385/1/012035](https://doi.org/10.1088/1742-6596/2385/1/012035).
- [21] Gabor Csardi et al. *igraph: Network Analysis and Visualization in R*. InterJournal, 2023. URL: <https://igraph.org>.
- [22] Leon Danon et al. “Comparing community structure identification”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2005.09 (Sept. 2005), p. 9008.
- [23] Abdelhani Diboune et al. “A comprehensive survey on community detection methods and applications in complex information networks”. In: *Social Network Analysis and Mining* 14.1 (2024), p. 93. DOI: [10.1007/s13278-024-01246-5](https://doi.org/10.1007/s13278-024-01246-5). URL: <https://doi.org/10.1007/s13278-024-01246-5>.
- [24] Duy Hieu Do and Thi Ha Duong Phan. *An improvement on the Louvain algorithm using random walks*. 2024. arXiv: 2403.08313 [cs.SI]. URL: <https://arxiv.org/abs/2403.08313>.
- [25] Daniel Edler, Anton Holmgren, and Martin Rosvall. *The MapEquation software package*. <https://mapequation.org>. 2023.
- [26] Henry Etzkowitz and Loet Leydesdorff. “The dynamics of innovation: from National Systems and “Mode 2” to a Triple Helix of university–industry–government relations”. In: *Research Policy* 29.2 (2000), pp. 109–123. ISSN: 0048-7333. DOI: [https://doi.org/10.1016/S0048-7333\(99\)00055-4](https://doi.org/10.1016/S0048-7333(99)00055-4). URL: <https://www.sciencedirect.com/science/article/pii/S0048733399000554>.
- [27] European Commission. *Call HORIZON-JTI-CLEANH2-2022-2 - Horizon Europe Joint Technology Initiative on Clean Hydrogen*. https://www.clean-hydrogen.europa.eu/call-proposals-2022_en. Accessed: 2024-08-26. URL: [%7Bhttps://www.clean-hydrogen.europa.eu/call-proposals-2022_en%7D](https://www.clean-hydrogen.europa.eu/call-proposals-2022_en%7D).
- [28] European Commission. *Commission Recommendation of 29 October 2009 on the use of the International Standard Classification of Occupations (ISCO-08)*. Official Journal of the European Union. 2009.

- [29] European Commission. *CORDIS: EU Research Projects under Horizon 2020*. URL: <https://data.europa.eu/data/datasets/cordish2020projects> Accessed: Sept. 2024. URL: %5Curl%7Bhttps://data.europa.eu/data/datasets/cordish2020projects%7D.
- [30] European Commission. *CORDIS: EU Research Projects under Horizon Europe 2021-2027*. URL: <https://data.europa.eu/data/datasets/cordis-eu-research-projects-under-horizon-europe-2021-2027> Accessed: Sept. 2024. URL: %5Curl%7Bhttps://data.europa.eu/data/datasets/cordis-eu-research-projects-under-horizon-europe-2021-2027%7D.
- [31] European Commission. *CORDIS: EU Research Results*. URL: <https://cordis.europa.eu> Accessed: Sept. 2024. URL: %5Curl%7Bhttps://cordis.europa.eu%7D.
- [32] European Commission. *EU Funding & Tenders Portal: Calls for Proposals*. Accessed: 2023-10-20. 2023. URL: <https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/calls-for-proposals>.
- [33] European Commission. *European Innovation Scoreboard 2023*. European Commission, Directorate-General for Internal Market, Industry, Entrepreneurship and SMEs. Accessed: 2024-10-22. 2023. URL: <https://ec.europa.eu/innovation-scoreboard>.
- [34] European Commission. *Horizon Europe Performance - Programme Performance Statements*. Accessed: 2023-10-20. 2023. URL: https://commission.europa.eu/strategy-and-policy/eu-budget/performance-and-reporting/programme-performance-statements/horizon-europe-performance_en.
- [35] European Commission. *North Adriatic Hydrogen Valley - project website*. <https://cordis.europa.eu/project/id/101111927>. Accessed: 2024-08-26. URL: <https://cordis.europa.eu/project/id/101111927>.
- [36] European Commission. *Open Research Europe*. <https://open-research-europe.ec.europa.eu>. Accessed: 2024-10-29. 2024.
- [37] European Commission. *Technology Readiness Levels (TRL): Extract from the Horizon 2020 Work Programme 2014-2015*. Accessed: 20-Oct-2024. 2014. URL: https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf.
- [38] Bojan Evkoski, I Mozetič, and Petra Kralj Novak. "Community evolution with ensemble louvain". In: *Complex networks* (2021), pp. 58–60.
- [39] Bojan Evkoski et al. "Community evolution in retweet networks". In: *PLOS ONE* 16.9 (Sept. 2021). Ed. by Chantal Cherifi, e0256175. DOI: 10.1371/journal.pone.0256175.

- [40] Santo Fortunato and Marc Barthelemy. "Resolution limit in community detection". In: *Proceedings of the national academy of sciences* 104.1 (2007), pp. 36–41.
- [41] Santo Fortunato and Darko Hric. "Community detection in networks: A user guide". In: *Physics Reports* 659 (2016), pp. 1–44.
- [42] Santo Fortunato and Mark EJ Newman. "20 years of network community detection". In: *Nature Physics* 18.8 (2022), pp. 848–850.
- [43] M. Frankowska and K. Cheba. "Exploring the Research Landscape of Hydrogen Valleys: A Bibliometric Analysis". In: *Journal of Sustainable Development of Transport and Logistics* 8.2 (2023), pp. 348–359. DOI: 10.14254/jsdtl.2023.8-2.27.
- [44] Benjamin H. Good, Yves-Alexandre de Montjoye, and Aaron Clauset. "Performance of modularity maximization in practical contexts". In: *Phys. Rev. E* 81 (4 Apr. 2010), p. 046106. DOI: 10.1103/PhysRevE.81.046106. URL: <https://link.aps.org/doi/10.1103/PhysRevE.81.046106>.
- [45] Economist Impact. *Open Innovation*. <https://impact.economist.com/projects/open-innovation>. Accessed: 2024-10-14. 2024.
- [46] Economist Impact. *Open Innovation Briefing Paper*. <https://impact.economist.com/projects/open-innovation/Open%20Innovation%20Briefing%20Paper.pdf>. Accessed: 2024-10-14. 2024.
- [47] Di Jin et al. "A survey of community detection approaches: From statistical modeling to deep learning". In: *IEEE Transactions on Knowledge and Data Engineering* 35.2 (2021), pp. 1149–1170.
- [48] Faiza Riaz Khawaja et al. "Exploring community detection methods and their diverse applications in complex networks: a comprehensive review". In: *Social Network Analysis and Mining* 14.1 (June 2024), p. 115. ISSN: 1869-5469. DOI: 10.1007/s13278-024-01274-1. URL: <https://doi.org/10.1007/s13278-024-01274-1>.
- [49] Yi-Xiu Kong et al. "k-core: Theories and applications". In: *Physics Reports* 832 (2019). k-core: Theories and Applications, pp. 1–32. ISSN: 0370-1573. DOI: <https://doi.org/10.1016/j.physrep.2019.10.004>. URL: <https://www.sciencedirect.com/science/article/pii/S037015731930328X>.
- [50] Andrea Lancichinetti and Santo Fortunato. "Consensus clustering in complex networks". In: *Scientific reports* 2.1 (2012), p. 336.
- [51] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. "Benchmark graphs for testing community detection algorithms". In: *Physical review E* 78.4 (2008), p. 046110.

- [52] Clement Lee and Darren J Wilkinson. “A review of stochastic block models and extensions for graph clustering”. In: *Applied Network Science* 4.1 (2019), pp. 1–50.
- [53] A. Majka et al. “Hydrogen Valley as a Hub for Technological Cooperation Between Science, Business, Local Government and NGOs: An Overview of Approaches in Europe”. In: *Torun International Studies* 1.17 (2023), pp. 5–15. DOI: [10.12775/TIS.2023.00](https://doi.org/10.12775/TIS.2023.00).
- [54] G. Menardi and D. De Stefano. “Density-based clustering of social networks”. In: *arXiv preprint arXiv:2101.08334* (2021).
- [55] Barend Mons et al. “The FAIR Principles: First Generation Implementation Choices and Challenges”. In: *Data Intelligence* 2.1-2 (Jan. 2020), pp. 1–9. ISSN: 2641-435X. DOI: [10.1162/dint_e_00023](https://doi.org/10.1162/dint_e_00023). eprint: https://direct.mit.edu/dint/article-pdf/2/1-2/1/1893425/dint_e_00023.pdf. URL: https://doi.org/10.1162/dint%5C_e%5C_00023.
- [56] Atefeh Moradan et al. “Ucode: Unified community detection with graph convolutional networks”. In: *Machine Learning* 112.12 (2023), pp. 5057–5080.
- [57] Fabio Morea. <https://github.com/fabio-morea/communities>. Version v 1.0. Aug. 2024. DOI: [10.5281/zenodo.13594210](https://doi.org/10.5281/zenodo.13594210).
- [58] Fabio Morea. *Horizon Projects Network Dataset*. Dataset, DOI: <https://doi.org/10.5281/zenodo.13765372>. 2024. DOI: [10.5281/zenodo.13765372](https://doi.org/10.5281/zenodo.13765372). URL: <https://zenodo.org/record/13765372>.
- [59] Fabio Morea and Domenico De Stefano. “Innovation Patterns within a Regional Economy through Consensus Community Detection on Labour Market Network”. In: *Proceedings of the Statistics and Data Science Conference* (2023). URL: <https://arts.units.it/handle/11368/3046559>.
- [60] Fabio Morea, Alberto Soraci, and Domenico De Stefano. *Mapping leadership and communities in EU-funded research through network analysis*. version 1; peer review: awaiting peer review. 2024. DOI: [10.12688/openreseurope.18544.1](https://doi.org/10.12688/openreseurope.18544.1). URL: <https://doi.org/10.12688/openreseurope.18544.1>.
- [61] Fabio Morea and Domenico De Stefano. *Beyond One Solution: The Case for a Comprehensive Exploration of Solution Space in Community Detection*. 2024. arXiv: [2410.19495](https://arxiv.org/abs/2410.19495) [cs.SI]. URL: <https://arxiv.org/abs/2410.19495>.
- [62] Fabio Morea and Domenico De Stefano. *Enhancing Stability and Assessing Uncertainty in Community Detection through a Consensus-based Approach*. Preprint, DOI: <https://doi.org/10.48550/arXiv.2408.02959>. 2024. arXiv: [2408.02959](https://arxiv.org/abs/2408.02959) [cs.SI]. URL: <https://arxiv.org/abs/2408.02959>.

- [63] M. E. J. Newman and M. Girvan. "Finding and evaluating community structure in networks". In: *Phys. Rev. E* 69 (2 2004), p. 026113.
- [64] Mark EJ Newman. "Finding community structure in networks using the eigenvectors of matrices". In: *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 74.3 (2006), p. 036104.
- [65] Mark EJ Newman and Michelle Girvan. "Finding and evaluating community structure in networks". In: *Physical review E* 69.2 (2004), p. 026113.
- [66] OpenAIRE. *Zenodo*. 2013. DOI: 10.25495/7GXX-RD71. URL: <https://about.zenodo.org/>.
- [67] Gergely Palla et al. "Uncovering the overlapping community structure of complex networks in nature and society". In: *nature* 435.7043 (2005), pp. 814–818.
- [68] J. Park, I.B. Wood, and E. et al. Jing. "Global labor flow network reveals the hierarchical organization and dynamics of geo-industrial clusters". In: *Nature Communications* 10 (2019), p. 3449. DOI: 10.1038/s41467-019-11380-.
- [69] Clean Hydrogen Partnership. *Clean Hydrogen Partnership*. <https://www.clean-hydrogen.europa.eu>. Accessed: 2024-10-20.
- [70] Clean Hydrogen Partnership. *Hydrogen Valleys: Insights into the emerging hydrogen economies around the world*. https://www.clean-hydrogen.europa.eu/system/files/2021-06/20210527_Hydrogen_Valleys_final_ONLINE.pdf. 2021.
- [71] Paolo Perlasca et al. "Multi-resolution visualization and analysis of biomolecular networks through hierarchical community detection and web-based graphical tools". In: *PLOS ONE* 15.12 (Dec. 2020), pp. 1–28. DOI: 10.1371/journal.pone.0244241. URL: <https://doi.org/10.1371/journal.pone.0244241>.
- [72] Mario Petrollese et al. "Techno-economic Assessment of Green Hydrogen Valley Providing Multiple End-users". In: *International Journal of Hydrogen Energy* 47.57 (2022), pp. 348–359.
- [73] Alexander Ponomarenko, Leonidas Pitsoulis, and Marat Shamshetdinov. "Overlapping community detection in networks based on link partitioning and partitioning around medoids". In: *PLOS ONE* 16.8 (Aug. 2021), pp. 1–43. DOI: 10.1371/journal.pone.0255717. URL: <https://doi.org/10.1371/journal.pone.0255717>.

- [74] Pascal Pons and Matthieu Latapy. "Computing communities in large networks using random walks". In: *Computer and Information Sciences-ISCIS 2005: 20th International Symposium, Istanbul, Turkey, October 26-28, 2005. Proceedings 20*. Springer. 2005, pp. 284–293.
- [75] M. E. Porter. "Location, competition, and economic development: Local clusters in a global economy". In: *Economic Development Quarterly* 14.1 (2000), pp. 15–34.
- [76] Valérie Poulin and François Theberge. "Ensemble clustering for graphs: comparisons and applications". In: *E. Appl Netw Sci* 4, 51 (2019). DOI: <https://doi.org/10.1007/s41109-019-0162-z>.
- [77] Publications Office of the European Union. *EuroSciVoc - European Science Vocabulary*. <https://cordis.europa.eu/article/id/430940-euroscivoc>. Accessed: 2024-09-24.
- [78] R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing, 2022. URL: <https://www.R-project.org>.
- [79] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. "Near linear time algorithm to detect community structures in large-scale networks". In: *Physical review E* 76.3 (2007), p. 036106.
- [80] Bruce S. Rawlings and Simon M. Reader. "What Is Innovation?: A Review of Definitions, Approaches, and Key Questions in Human and Non-Human Innovation". In: *The Oxford Handbook of Cultural Evolution*. Oxford University Press, 2024. ISBN: 9780198869252. DOI: 10.1093/oxfordhb/9780198869252.013.11. URL: <https://doi.org/10.1093/oxfordhb/9780198869252.013.11>.
- [81] André Luis Rossoni, Eduardo Pinheiro Gondim de Vasconcellos, and Renata Luiza de Castilho Rossoni. "Barriers and facilitators of university-industry collaboration for research, development and innovation: a systematic review". In: *Management Review Quarterly* 74.3 (2024), pp. 1841–1877. ISSN: 2198-1639. DOI: 10.1007/s11301-023-00349-1. URL: <https://doi.org/10.1007/s11301-023-00349-1>.
- [82] M. Rosvall, D. Axelsson, and C. T. Bergstrom. "The map equation". In: *The European Physical Journal Special Topics* 178.1 (Nov. 2009), pp. 13–23. DOI: 10.1140/epjst/e2010-01179-1. URL: <https://doi.org/10.1140%2Fepjst%2Fe2010-01179-1>.
- [83] Martin Rosvall and Carl T Bergstrom. "An information-theoretic framework for resolving community structure in complex networks". In: *Proceedings of the national academy of sciences* 104.18 (2007), pp. 7327–7331.

- [84] Martin Rosvall and Carl T. Bergstrom. “Maps of random walks on complex networks reveal community structure”. In: *Proceedings of the National Academy of Sciences* 105.4 (2008), pp. 1118–1123. DOI: [10.1073/pnas.0706851105](https://doi.org/10.1073/pnas.0706851105). eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.0706851105>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.0706851105>.
- [85] Subhajit Sahu. “Addressing Internally-Disconnected Communities in Leiden and Louvain Community Detection Algorithms”. In: *arXiv preprint arXiv:2402.11454* (2024).
- [86] Joseph A. Schumpeter. *The Theory of Economic Development: An Inquiry into Profits, Capital, Credit, Interest, and the Business Cycle*. Cambridge, MA: Harvard University Press, 1934.
- [87] Fernanda Rosalina da Silva Meireles, Ana Cláudia Azevedo, and João Maurício Gama Boaventura. “Open innovation and collaboration: A systematic literature review”. In: *Journal of Engineering and Technology Management* 65 (2022), p. 101702. ISSN: 0923-4748. DOI: <https://doi.org/10.1016/j.jengtecman.2022.101702>. URL: <https://www.sciencedirect.com/science/article/pii/S0923474822000327>.
- [88] Blaž Škrlj, Jan Kralj, and Nada Lavrač. “Embedding-based Silhouette community detection”. In: *Machine Learning* 109 (2020), pp. 2161–2193.
- [89] Daniela Stoltenberg, Daniel Maier, and Annie Waldherr. “Community detection in civil society online networks: Theoretical guide and empirical assessment”. In: *Social Networks* 59 (2019), pp. 120–133. ISSN: 0378-8733. DOI: <https://doi.org/10.1016/j.socnet.2019.07.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0378873318301138>.
- [90] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. “From Louvain to Leiden: guaranteeing well-connected communities”. In: *Scientific reports* 9.1 (2019), p. 5233.
- [91] U. Weichenhain et al. *GOING GLOBAL: An update on Hydrogen Valleys and their role in the new hydrogen economy*. Clean Hydrogen Partnership, 2021. ISBN: 978-92-9246-394-6. URL: https://www.clean-hydrogen.europa.eu/system/files/2021-06/20210527_Hydrogen_Valleys_final_ONLINE.pdf.
- [92] Mark D. Wilkinson et al. “The FAIR Guiding Principles for scientific data management and stewardship”. In: *Scientific Data* 3.1 (2016), p. 160018. ISSN: 2052-4463. DOI: [10.1038/sdata.2016.18](https://doi.org/10.1038/sdata.2016.18). URL: <https://doi.org/10.1038/sdata.2016.18>.
- [93] Wayne W Zachary. “An information flow model for conflict and fission in small groups”. In: *Journal of Anthropological Research* 33.4 (1977), pp. 452–473.